On the Relative Expressive Power of Contextual Grammars with Maximal and Depth-First Derivations

Lakshmanan Kuppusamy[1,⋆] and Kamala Krithivasan[2]

[1] ALPAGE, INRIA-Rocquencourt
Le Chesnay - 78153, France
klakshma@vit.ac.in

[2] Department of Computer Science and Engineering
Indian Institute of Technology Madras
Chennai - 600 036, India
kamala@iitm.ac.in

Abstract. In the recent years, several new classes of contextual grammars have been introduced to give an appropriate model description to natural languages. With this aim, some new families of contextual languages have been introduced based on maximal and depth-first conditions and analyzed in the framework of so-called mildly context sensitive languages. However, the relationship among these families of languages have not yet been analyzed in detail. In this paper, we investigate the relationship between the families of languages whose grammars are based on maximal and depth-first conditions. We prove an interesting result that all these families of languages are incomparable to each other, but they are not disjoint.

Keywords: internal contextual grammars, maximal, depth-first, incomparable.

1 Introduction

Contextual grammars produce languages starting from a finite set of *axioms* and adjoining *contexts*, iteratively, according to the *selector* present in the current sentential form. As introduced in [15], if the contexts are adjoined at the ends of the strings, the grammar is called *external*. *Internal* contextual grammars were introduced by Păun and Nguyen in 1980 [20], where the contexts are adjoined to the selector strings which appear as substrings of the derived string. The main motivation for introducing contextual grammars was to obtain languages that are more appropriate from natural languages point of view. In fact, the class of languages should (i) contain basic non-context-free languages, (ii) be parsable in polynomial time (iii) contain *semilinear* languages only, and these three properties together define the so-called *mildly context sensitive* (MCS) formalisms and languages, as introduced by A.K. Joshi in 1985 [5].

⋆ Currently on leave from parent Institution VIT University, Vellore - 632 014, India.

M. Leucker and C. Morgan (Eds.): ICTAC 2009, LNCS 5684, pp. 246–260, 2009.

When contextual grammars are analyzed from the perspective of MCS formalisms, the basic classes, external and internal contextual languages fail to contain some desirable non-context-free languages. Further, they contain non-semilinear languages too [4],[6]. Also, at present only exponential time algorithms are known for the membership problem of internal contextual grammars [2] and whether it can be solved in polynomial time algorithm remains open [8]. Therefore, some attempts have been made in the last decade or so to introduce variants of contextual grammars by restricting the selector chosen in the derivation, to obtain certain specific classes of contextual languages which satisfy the above said MCS properties. The first such main variant was *depth-first* contextual grammars [18] where the main catch is to track the previously adjoined contexts in the selector. Though this idea might be useful while parsing (especially for backtracking), these grammars fail to generate one of the basic non-context-free languages, like *multiple agreement:* $\{a^n b^n c^n \mid n \geq 1\}$. So, other new classes of grammars have been introduced, for instance, *maximal* contextual grammars [17]. Though they generate the basic non-context-free languages, they also generate non-semilinear languages [16]. Besides, in [2], it was proved that these maximal and internal contextual grammars can be transformed into equivalent *dynamic range concatenation grammars*, an extended formalism of *range concatenation grammars* [1]). However, parsing dynamic range concatenation grammar allows exponential time complexity and thus this strategy is not useful.

Further, in [11], a variant namely *maximal depth-first* grammars have been introduced, by combining the maximal and depth-first conditions. Like maximal grammars, the family of languages generated by these grammars contain non-context-free languages, but their membership and semilinear problems have been left open. Later in [12], two variants, namely *end-marked maximal depth-first* and *inner end-marked maximal depth-first* grammars have been introduced with the aim to solve the membership problem and semilinearity issue for maximal depth-first derivation. In [7], Ilie considered a new variant called *maximal local*. Ilie showed that the languages generated by maximal local grammars with regular selectors contain basic non-context-free languages and the membership problem for these languages is solvable in polynomial time. But the question of semilinearity was left open for these languages and in [11], a restricted variant of maximal local contextual grammars, called *absorbing right context grammar* has been introduced in order to solve the semilinear problem of maximal local.

Many of these variants were obtained by refining the previous variants (i.e., imposing further restrictions in the existing variants) with the hope that they could clear the failed properties of MCS and at the same time the properties which are shown to satisfy are also preserved. Out of all these variants discussed above, the classes of languages generated by maximal local, absorbing right context, inner end-marked depth-first grammars with regular selectors were shown to satisfy the properties of MCS languages [9],[11],[12]. Since all the above variants have been introduced with a single aim to satisfy the properties of MCS languages (and thus to give a model description for natural languages from the domain of contextual grammars), the relative expressive power of these variants have not been discussed so far. Our motivation in this paper is to analyze the expressive power of these grammars.

When several classes of grammars originate from one grammar, it would be interesting to analyze their power of generating languages and to form the hierarchical structures with the results we obtain. When such hierarchical order is not possible between the families of languages, they become incomparable. In this paper, we analyze the generative power of the above mentioned variants of the internal contextual grammars with regular selectors. We prove that all these families of the above said variants are incomparable to one another. Also, we prove that they are not disjoint as there are common languages that are shared by these families of languages.

As a word of caution we would like to mention that so far no unanimous definition of MCS has been agreed to. For example, the semilinear property is considered to be too strong and is replaced by a weaker property, *constant growth property.* Also non-compliance of these mentioned properties does not rule out a formalism being useful, for example, "back-end" general formalisms like range concatenation grammars [1] or abstract categorial grammars [19]. Also, several other variants of contextual grammars have been introduced and analyzed from the perspective of formal languages. However, we do not discuss them here as it is out of scope of this paper and we refer to the monograph [21] for more variants.

2 Preliminaries

We assume the readers are familiar with the basic formal language theory notions. We refer to [22] for more details on formal language theory. We now present the definition of a few classes of contextual grammars considered in this paper.

An *internal contextual grammar* is $G = (V, A, (S_1, C_1), \ldots, (S_m, C_m)), m \geq 1$, where V is an alphabet, $A \subseteq V^*$ is a finite set called *axioms*, $S_j \subseteq V^*$, $1 \leq j \leq m$, are the sets of *selectors* or *choice*, and $C_j \subseteq V^* \times V^*, C_j$ finite, $1 \leq j \leq m$, are the sets of *contexts* associated with the selector S_j. The usual *derivation* in the *internal mode* is defined as $x \Longrightarrow_{in} y$ iff $x = x_1x_2x_3,\ y = x_1ux_2vx_3,$ for $x_1, x_2, x_3 \in V^*,\ x_2 \in S_j, (u, v) \in C_j,\ 1 \leq j \leq m.$

Given an internal contextual grammar G as above, the maximal and depth-first derivations are given as below. In maximal mode (denoted by *max*), at each derivation, the chosen selector $x_2 \in S_i$, for the next derivation should be of maximal length than the other possible selectors $x_2' \in S_i$ (for the formal representation of maximal condition, refer the below condition (iii) alone). In depth-first mode (denoted by *df*), for every derivation, the selector for the next derivation must contain one of the contexts u or v which was adjoined in the previous derivation (for the formal representation of depth-first condition, refer below condition (ii) alone). Next, we define maximal depth-first grammar, obtained by combining maximality and depth-first conditions. More formally, given a contextual grammar G as above, a *maximal depth-first derivation* (denoted by *mdf*) in G is a derivation $w_1 \Longrightarrow_{mdf} w_2 \Longrightarrow_{mdf} \ldots \Longrightarrow_{mdf} w_n,\ n \geq 1,$ where

(i) $w_1 \in A,\ w_1 \Longrightarrow_{in} w_2$ (i.e., in the usual internal mode),

(ii) For each $i = 2, 3, \ldots, n-1$, if $w_{i-1} = z_1 z_2 z_3,\ w_i = z_1 u z_2 v z_3$ ((u, v) is the context adjoined to w_{i-1} in order to get w_i), then $w_i = x_1 x_2 x_3,\ w_{i+1} = x_1 s x_2 t x_3$, such that $x_2 \in S_j,\ (s, t) \in C_j$, for some $j,\ 1 \leq j \leq m$, and x_2 contains one of the contexts u or v as a substring (thus, satisfying the depth-first condition). Note that here the chosen next selector contains not any s or t occurred in the string, but the same s or t adjoined in the previous derivation step.

(iii) For each $i = 2, 3, \ldots, n-1$, if $w_i \Longrightarrow_{df} w_{i+1}$, then there will be no other derivation in G with $w_i \Longrightarrow_{df} w'_{i+1}$ such that $w_i = x'_1 x'_2 x'_3,\ x'_2 \in S_j$ and $|x'_2| > |x_2|$ where $x_2 \in S_j$ (note that the selector x_2 is of maximal length with respect to S_j only, and not with respect to all selectors).

Given a contextual grammar G, we next define the *local* mode in the following way. For $z \in A,\ z \Longrightarrow_{in} x$ such that $z = z_1 z_2 z_3,\ x = z_1 u z_2 v z_3,\ z_2 \in S_k, (u, v) \in C_k$, for $z_1, z_2, z_3 \in V^*,\ 1 \leq k \leq m$, then $x \Longrightarrow_{loc} y$ is called local with respect to $z \Longrightarrow x$, iff we have $u = u'u'',\ v = v'v'',\ u', u'', v', v'' \in V^*,\ y = z_1 u' s u'' z_2 v' t v'' z_3$, for $u'' z_2 v' \in S_j,\ (s, t) \in C_j,\ 1 \leq j \leq m$. That is, at each derivation, the contexts are introduced adjacent to the contexts (or to the side of the previous selector itself, when $u'' = \lambda = v'$) which were introduced in the previous derivation. Note that, at every derivation, the selector may expand on its left side or right side or both sides, but expands not more than the contexts introduced in the previous derivation step. Therefore, once a selector is chosen, that selector should be a subword for the selectors used in the further derivations (this point is often used in the proofs). When the maximality condition is included with this local variant, the grammar is said to be *maximal local* (denoted by *mloc*).

Now, we define a variant obtained by imposing further restriction to the above *mloc* grammar and we call it as *absorbing right contextual grammar* (denoted by *arc*) [11]. In this variant, the selector (say y_{i+1}) for the next derivation (step) is obtained by adjoining the first half v'_i of the current right context to the current selector y_i where $v_i = v'_i v''_i,\ |v'_i| = \lceil \frac{|v_i|}{2} \rceil,\ |v''| = \lfloor \frac{|v_i|}{2} \rfloor$. That is, $y_{i+1} = y_i v'_i,\ y_i \in S_j,\ y_i \in V^*, v'_i \in V^+$.

An *end-marked maximal depth-first* (denoted by *emdf*) contextual grammar [12] is a construct $G = (V, A, \{(S_1, C_1), \ldots, (S_m, C_m)\}), m \geq 1$, where $V, A, S_1, \ldots S_m$, are as mentioned in the definition of internal contextual grammar and $C_j \subseteq (V^+_{\{L,R\}} \times V^*) \cup (V^* \times V^+_{\{L,R\}})$, C_j finite, $1 \leq j \leq m$, are the set of contexts. The elements of C_j's are of the form $(u_L, v), (u_R, v), (u, v_L)$, and (u, v_R). The suffix L and R represents end marker (left and right) for the selector of the next derivation. u_L (or v_L) indicates the selector for the next derivation should start with u (or v), thus u (or v) is the left end of the next selector. Similarly, u_R (or v_R) indicates the selector for the next derivation should end with the context u (or v). Given such a grammar G, an *emdf* derivation in G is a derivation $w_1 \Longrightarrow_{emdf} w_2 \Longrightarrow_{emdf} \ldots \Longrightarrow_{emdf} w_n,\ n \geq 1$, where

– $w_1 \in A,\ w_1 \Longrightarrow w_2$ in the usual way,

– For each $i = 2, 3, \ldots, n-1$, if $w_{i-1} = z_1 z_2 z_3$, $w_i = z_1 u z_2 v z_3$, such that $z_2 \in S_k$, $1 \leq k \leq m$, then $w_i = x_1 x_2 x_3$, $w_{i+1} = x_1 s x_2 t x_3$, such that $x_2 \in S_j$, $1 \leq j \leq m$, and x_2 will be one of the following four cases:
  (i) $x_2 = u z_2'$, $u \neq \lambda$, if $(u_L, v) \in C_k$, with $z_2' \in V^*$ is of maximal (i.e., there exists no $z_2'' \in V^*$, such that $u z_2'' \in S_j$, with $|z_2''| > |z_2'|$).
  (ii) $x_2 = z_1' u$, $u \neq \lambda$, if $(u_R, v) \in C_k$, with $z_1' \in V^*$ is of maximal (i.e., there exists no $z_1'' \in V^*$, such that $z_1'' u \in S_j$, with $|z_1''| > |z_1'|$).
  (iii) $x_2 = z_2' v$, $v \neq \lambda$, if $(u, v_R) \in C_k$, with $z_2' \in V^*$ is of maximal (i.e., there exists no $z_2'' \in V^*$, such that $z_2'' v \in S_j$, with $|z_2''| > |z_2'|$).
  (iv) $x_2 = v z_3'$, $v \neq \lambda$, if $(u, v_L) \in C_k$, with $z_3' \in V^*$ is of maximal (i.e., there exists no $z_3'' \in V^*$, such that $v z_3'' \in S_j$, with $|z_2''| > |z_2'|$).

Now, we introduce the next variant. Given a *emdf* grammar G, we can define the *inner end-marked maximal depth-first grammar* (denoted by *iemdf*) by imposing the following changes in the grammar and in derivation.

– $C_j \subseteq (V_L^+ \times V^*) \cup (V^* \times V_R^+)$.
– As the elements of C_j's are of the form (u_L, v) and (u, v_R), the cases (ii) and (iv) discussed above are void and only the cases (i) and (iii) are valid.
– The selector for the next derivation should lie inside the contexts u and v which were adjoined in the previous derivation. More precisely, the next chosen selector cannot have both the adjoined contexts u and v, but it may contain the proper prefixes of v (if u is end-marked, i.e., u_L) or proper suffixes of u (if v is end-marked, i.e., v_R). Obviously the end-marked context is included in the next chosen selector in order to satisfy the depth-first and end-marked conditions. More formally, if u and v are the contexts adjoined to the selector, say z_2, then the next selector, say x_2, will be a strict subword of $u z_2 v$ and x_2 should either begin with u or end with v.

From the above definitions, we can see that the definition of each of the grammars is interlinked with the other and all the grammars share the maximality condition in common (except *arc*) and many grammars share the depth-first condition also (some grammars share this condition partially, like *mloc* and *arc*).

The language generated by a grammar G in the mode β, $\beta \in \{max, mdf, mloc, arc, emdf, iemdf\}$ is given by $L_\beta(G) = \{w \in V^* \mid x \Longrightarrow_\beta^* w,\ x \in A\}$, where $\Longrightarrow_\beta^*$ is the reflexive transitive closure of the relation $\Longrightarrow_\beta$. If all the sets of selectors $S_1, \ldots, S_m$ are in a family F of languages, then we say that the grammar G is with F choice. As usual, the family of languages for G working in $\beta \in \{max, mdf, mloc, arc, emdf, iemdf\}$ mode with F choice is given as $ICC_{max}(F), ICC_{mdf}(F), ICC_{mloc}(F), ICC_{arc}(F), ICC_{emdf}(F)$, and $ICC_{iemdf}(F)$, respectively. In this paper, we consider $F \in \{FIN, REG\}$.

The following assumption is made throughout this paper. We do not consider the empty contexts (λ, λ) here, but one-sided contexts of the form $(\lambda, v), (u, \lambda)$ are considered (but the λ context cannot be an end-marker). Also, the underlined symbols denote the newly inserted contexts and the word in between the two down arrows indicates the selector used for the next derivation. We call maximal length as maximal in many places for the sake of brevity. Also, we refer the selector for the next derivation as simply next selector in many occurrences.

3 Results

In this section, we discuss the generative power of the internal contextual grammars when we put different types of restrictions on the derivations such as $max, mloc, arc, mdf, emdf, iemdf$. Here, the generative power of a class of grammars deals with the limitation of the grammars in generating the languages (like what languages can or cannot be produced by these grammars). We aim to show that there are some languages which can be generated when putting one type of restriction on the derivation but they cannot be generated when some other types of restriction is imposed on the derivation. Also we aim to show that there are lanaguges which can be generated by all types of restricted derivations mentioned in the previous section.

Lemma 1. *$ICC_\alpha(FIN) \subset ICC_\alpha(REG)$, $\alpha \in \{max, mdf, mloc, arc, emdf, iemdf\}$.*

Proof. The relation $ICC_\alpha(FIN) \subseteq ICC_\alpha(REG)$ is obvious. The strict inclusion follows from the following result. Consider the crossed dependency language $L_1 = \{a^n b^m c^n d^m \mid n, m \geq 1\}$. This language cannot be generated by any of the above α grammars with finite choice since in order to increase the occurrences of a, c equally and b, d equally, the grammar needs regular selectors of the form $a^{k_1} b^+ c^{k_2}$ and $b^{k_3} c^+ d^{k_4}, k_1, k_2, k_3, k_4 \geq 0$, respectively. However, in previous papers ([7],[9],[11],[12],[17]), all these grammars were shown to generate L_1 with regular selectors. □

Lemma 2. *$L_2 = \{a, b\}^+ \in ICC_\alpha(REG)$, $\alpha \in \{max, mdf, mloc, arc, emdf, iemdf\}$.*

Proof. The language $L_2 = \{a, b\}^+$ can be generated by $G_\alpha = (\{a, b\}, \{a, b\}, (\{a, b\}, \{(a_L, \lambda), (b_L, \lambda)\}))$ for $\alpha \in \{max, mdf, emdf, iemdf\}$ (for max, mdf modes, there is no suffix L in the contexts). Any string $w = w_1 \ldots w_n \in L_2$ can be produced by starting from w_n, adding the context on the left, iteratively. For $\beta = \{mloc, arc\}$ modes, $G_\beta = (\{a, b\}, \{a, b\}, (\{a, b\}^+, \{(\lambda, a), (\lambda, b)\}))$. It is easy to see that $L(G_\beta) = L_2$. □

The above result shows that the language $\{a, b\}^+$ is included in all families of languages $ICC_\alpha(REG)$, $\alpha \in \{max, mdf, mloc, arc, emdf, iemdf\}$.

Lemma 3. *$ICC_\alpha(REG) - ICC_{max}(REG) \neq \emptyset$, $\alpha \in \{mdf, mloc, arc, emdf, iemdf\}$.*

Proof. The language $L_3 = \{a^n \mid n \geq 1\} \cup \{a^n b^n c^n \mid n \geq 1\}$ can be generated by the grammars

$$\begin{aligned} G_{mdf} &= (\{a, b, c\}, \{a, aa, abc\}, (aa, (a, \lambda)), (b^+c^+, (ab, c))). \\ G_{mloc} &= (\{a, b, c\}, \{a, aa, abc\}, (aa, (a, \lambda)), (b^+c, (ab, c))). \\ G_{arc} &= (\{a, b, c\}, \{a, aa, abc\}, (aa^+, (\lambda, a)), (b^+, (a, bc))). \\ G_{\{emdf, iemdf\}} &= (\{a, b, c\}, \{a, aa, abc\}, (aa, (a_L, \lambda)), (b^+c^+, (ab, c_R))). \end{aligned}$$

In order to get a better understanding on how the strings are generated using these grammars, we provide some details about the selectors used in the derivations.

To generate the strings of the form a^n, aa (or aa^+ for *arc* mode) is chosen as selector for all derivations and a is adjoined to the side of the selector. For strings of the other part of the language ($a^nb^nc^n$), the selector b^+c^+ covers the adjoined right context c in *mdf* mode. In *mloc* mode, we have $u_2' = a, u_2'' = b$ and $v_2' = \lambda, v_2'' = c$, at every derivation. In *arc* mode, every time the selector b^+ absorbs half of the right context b in bc. In *emdf* mode, whenever (a_L, λ) is introduced, the next selector starts with a (a is the left end of the selector aa) and whenever (ab, c_R) is introduced, the next selector ends with the adjoined right context c (c is the right end of the selector b^+c^+). In *iemdf* mode, the condition (u_L, v) or (u, v_R) is satisfied and the selector is inside the previously introduced contexts. In all modes, the selectors are chosen of maximal length.

However, the language L_3 is not in $ICC_{max}(REG)$. Assume that the language $L_3 \in ICC_{max}(REG)$ for a maximal grammar G_{max}. In order to generate the strings $a^n, n \geq 1$, we need a selector $a^k, k \geq 0$, with the context $(a^{i_1}, a^{i_2}), i_1+i_2 \geq 1$. Now, consider a string $a^pb^pc^p$ for a large $p \geq k$. As the context (a^{i_1}, a^{i_2}) can be applied to $a^pb^pc^p$ by choosing a subword a^k in a^p, we can produce strings of the form $a^{p+i_1+i_2}b^pc^p \notin L_3$. A contradiction. □

The following result is the counterpart for the above lemma.

Lemma 4. *$ICC_{max}(REG) - ICC_\beta(REG) \neq \emptyset$, $\beta \in \{mdf, mloc, arc, emdf, iemdf\}$.*

Proof. Consider the language $L_4 = \{a^ncb^na^mcb^m \mid n, m \geq 0\}$. It is in $ICC_{max}(FIN)$, because this language can be generated by the grammar $G_{max} = (\{a, b, c\}, cc, (c, (a, b)))$. By Lemma 1, $L_4 \in ICC_{max}(REG)$.

However, $L_4 \notin ICC_\beta(REG)$ for the above β. Assume that $L_4 \in ICC_\beta(REG)$ for any grammar $G_\beta = (\{a, b, c\}, A, (S_1, C_1), \ldots, (S_r, C_r))$. First, we give the proof for the case $\beta = mdf$. As axiom is also present in the language, the axiom A must have a word of the form $a^icb^ia^jcb^j, i, j \geq 0$, and a context of the form $(a^k, b^k), k \geq 1$, is adjoined to such a word, then either the number of occurrences of a and b around the first c, or the number of occurrences of a and b around the second c is increased. Assume that the occurrences of a and b around the first c is increased equally (the case of a and b increased equally around the second c is symmetric). Therefore, we have $a^icb^ia^jcb^j \Longrightarrow a^{i_1}\underline{a^k}a^{i_2}cb^{i_3}\underline{b^k}b^{i_4}a^jcb^j$ for $i_1+i_2 = i_3 + i_4 = i$. The derivation must continue using a selector which covers at least one of the contexts a^k or b^k. Continuing the derivation in this fashion, at some point of time, we have to increase the number of occurrences of a and b around the second c. In such a case, we have to use a context of the form $(a^p, b^p), p \geq 1$, and the selector should contain the subword b^k which was introduced in the previous derivation. As a^p is a left context, it cannot be added to the right side of b^k and so a^p should be adjoined to the left of b^k (but not necessarily immediate left). Then, we will have unequal number of a and b around the second c, which results in a word not in L_4. Other possibilities of derivations also lead to generation of strings not in the language. Therefore, $L(G_{mdf}) = L_4$ is impossible. For $\beta = emdf$ mode, as the definition is based on depth-first concept, the above argument about the context and selector are applicable. Continuing in that line, we have the context (a^k, b^k) is end-marked. Therefore, we have either $a^k_{L,R}$ or $b^k_{L,R}$. Obviously, a^k_R and b^k_L are failed to increase the occurrences of a and b around the second c. If a^k_L is the case, the occurrences of a around the second c cannot be increased and if b^k_R is the case, the occurrences of b around the second c cannot be increased, thus unequal occurrences a and b is generated. It is not hard to come-up with a similar argument to prove that the language cannot be generated by *iemdf* grammars. Now, let us take $\beta = mloc$. By definition of the grammar, every time the contexts are introduced adjacent to the previously introduced contexts or to the previously used selector, pumping equal number of a and b is possible only on one part of the language. Otherwise, we can derive a word which is not in the language using a similar technique as above.

Finally, let us consider the case for $\beta = arc$. From the language, it is obvious that no selector can have both c as a subword. Otherwise, b and a cannot be increased in between the two c. Since the selector accumulates only on its right side in this mode, if we use a selector contains the second c as a subword in the axiom, then we cannot pump equal occurrences of a and b around the first c. On the other hand, if we choose a selector which contains the first c as a subword, then a^*cb^+ will be a selector for further derivations. However, from this selector, we can increase the occurrences of a only in the second part, thus unequal number of a and b around the second c is generated. A contradiction. □

From Lemma 2, 3 and 4, we have the following theorem.

Theorem 1. *$ICC_{max}(REG)$ is incomparable with the families $ICC_\alpha(REG)$, for $\alpha \in \{mdf, mloc, arc, emdf, iemdf\}$, but not disjoint.*

Lemma 5. *$ICC_\alpha(REG) - ICC_{mdf}(REG) \neq \emptyset$, $\alpha \in \{mloc, arc, emdf, iemdf\}$.*

Proof. Consider the language $L_5 = \{a^n cb^n \mid n \geq 1\} \cup \{a^n \mid n \geq 1\}$. This language is in the family $ICC_\alpha(REG)$ for the above α. Because this language can be generated by the grammar $G_{arc} = (\{a, b, c\}, \{acb, a, aa\}, (aa^+, (\lambda, a)), (cb^+, (a, b)))$, in *arc* mode. For maximal local mode, the grammar $G_{mloc} = (\{a, b, c\}, \{acb, a, aa\}, (aa, (\lambda, a)), (acb, (a, b)))$ generates L_5. Note that, the selector aa cannot be used in the subword a^+cb^+ since once a selector is chosen in this mode, it will always be a subword to the further subwords. For *emdf* and *iemdf*, the grammar $G_5 = (\{a, b, c\}, \{acb, a, aa\}, (aa^+, (\lambda, a_R)), (acb^+, (a, b_R)))$ generates L_5.

However, $L_5 \notin ICC_{mdf}(REG)$. On contrary, let us assume that $L_5 \in ICC_{mdf}(REG)$ for a *mdf* grammar G_{mdf}. In order to generate the strings of the first part, we need a context of the form $(a^m, b^m), m \geq 1$. In order to obtain words of the form a^n for a large n, we need a context $(a^i, a^j), i+j \geq 1$, associated with the selector $a^k, k \geq 1$. Assume a word $a^{m+r}cb^{m+r}$ in the language where $m+r \geq k$. Also, assume that this word is derived from $a^rcb^r, r \geq 1$ by adjoining the context (a^m, b^m). The selector for the next derivation should contain one of the contexts a^m or b^m. Now we can use the selector a^k and obtain a word $a^{i+j+m+r}cb^{m+r} \notin L_5$. A contradiction. □

Lemma 6. *$ICC_\alpha(REG) - ICC_\beta(REG) \neq \emptyset$, $\alpha \in \{mloc, mdf, emdf\}$, $\beta \in \{arc, iemdf\}$.*

Proof. Consider the marked mirror image language $L_6 = \{wcw^r \mid w \in \{a,b\}^*\}$. This language can be generated by the grammars

$$\begin{aligned} G_{mloc} &= (\{a,b,c\}, c, (c, \{(a,a),(b,b)\})). \\ G_{mdf} &= (\{a,b,c\}, c, (\{w'cw'' \mid w', w'' \in \{a,b\}^*\}, \{(a,a),(b,b)\})). \\ G_{emdf} &= (\{a,b,c\}, c, (\{w'cw'' \mid w', w'' \in \{a,b\}^*\}, \{(a_L,a),(b_L,b)\})). \end{aligned}$$

However, this language does not belong to $ICC_{arc}(REG), ICC_{iemdf}(REG)$. Because, for any type of grammar, generating the strings of the form wcw^r is possible only when the context of the form $(a^i, a^i), i \geq 1$ or $(b^j, b^j), j \geq 1$, is adjoined to the selector c in each derivation, or when the above contexts are adjoined to the selector $w'cw''$, $w', w'' \in \{a,b\}^*$ and the selector $w'cw''$ is of maximal length. So, starting from c, either the selector c should absorb both right and left context or should not absorb any context. In *arc* grammars, as the selector absorbs the right context only, we cannot generate the language L_6 or otherwise, we can generate words which are not in L_6. In *iemdf* mode, though the selector can absorb right and left contexts, it is not permitted to absorb both contexts at a time since the chosen selector for the next derivation should be inside the adjoined contexts. Therefore, choosing a selector $w'cw''$ of maximal length is not possible. □

From Lemma 2, 5 and 6, we have the following theorem.

Theorem 2. *$ICC_{mdf}(REG)$ is incomparable with $ICC_{iemdf}(REG)$, but not disjoint.*

Lemma 7. *$ICC_\alpha(REG) - ICC_\beta(REG) \neq \emptyset$, $\alpha \in \{arc, iemdf\}$, $\beta \in \{mloc, mdf, emdf\}$.*

Proof. Consider the non-marked duplication language $L_7 = \{ww \mid w \in \{a,b\}^*\}$. This language can be generated by the grammars

$$\begin{aligned} G_{arc} &= (\{a,b\}, \lambda, (\{w' \mid w' \in \{a,b\}^*\}, \{(a,a),(b,b)\})), \\ G_{iemdf} &= (\{a,b\}, \lambda, (\{w' \mid w' \in \{a,b\}^*\}, \{(a,a_R),(b,b_R)\})). \end{aligned}$$

In *arc* mode, starting with the initial selector λ, it accumulates the right context a or b every time. In *iemdf* mode, every time, the selector for the next derivation is chosen inside the adjoined contexts (but right context is included for meeting the depth-first condition) and of maximal length. A sample derivation in α mode $\alpha \in \{arc, iemdf\}$ is given as

$$\lambda \Longrightarrow_\alpha \underline{w_1}^\downarrow \underline{w_1}^\downarrow \Longrightarrow_\alpha w_1\underline{w_2}^\downarrow w_1\underline{w_2}^\downarrow \Longrightarrow_\alpha w_1w_2\underline{w_3}^\downarrow w_1w_2\underline{w_3}^\downarrow \Longrightarrow_\alpha^* ww.$$

However L_7 does not belong to $ICC_\beta(REG)$ for the above β. In order to generate the strings of the form ww, at each derivation, from the derived string

$w'' \in L_7$, the context (x, x), $x \in \{a, b\}^+$ is adjoined at the beginning of w''(left context x) and at the center of w''(right context x) or at the center of w''(left context x) and at the end of w''(right context x). This implies, the chosen selector should expand only at one side from the center. Assume that $w' \in L_7$ is derived from w'' in such a way. Then, $w' = z\underline{x}z\underline{x}$ or $w' = \underline{x}z\underline{x}z$, for $z \in \{a, b\}^*$, $x \in \{a, b\}^+$ is the context adjoined and $w'' = zz$. For $\beta = mloc$ mode, at each derivation, the contexts are adjoined to the side of previously adjoined contexts and the selector (which is over $\{a, b\}$) is chosen of maximal length, from w' we can derive $z\underline{y}xzx\underline{y} \notin L_7$ or $\underline{y}xzx\underline{y}z \notin L_7$, where y is the context adjoined (which should be near the last adjoined context x). Next, we assume $\beta = emdf$. In *emdf* mode, we have the contexts are end-marked, thus $(x_{\{L,R\}}, x)$ or $(x, x_{\{L,R\}})$ is the case. If (x_R, x) is the case, we have $w' = z\underline{x_R}z\underline{x}$ or $\underline{x_R}z\underline{x}z$ and from w' we obtain, $w' \Longrightarrow \underline{y_R}zx\underline{y}zx$ or $w' \Longrightarrow \underline{y_R}x\underline{y}zxz \notin L_7$ for the adjoined context y. Though $\underline{y_R}zx\underline{y}zx \in L_7$, in the next derivation while adjoining another context (y'_R, y'), we would have $\underline{y'_R}y\underline{y'}zxyzx \notin L_7$. If (x_L, x) is the case, we have $w' = z\underline{x_L}zx$ or $w' = \underline{x_L}z\underline{x}z$ and from w', we obtain $w' \Longrightarrow z\underline{y_L}xzx\underline{y} \notin L_7$ or $w' \Longrightarrow \underline{y_L}xyzxz \notin L_7$, where is the adjoined context. For the other case $(x, x_{\{L,R\}})$, a similar proof can be given. Note that it is look like L_7 can be generated in *emdf* mode (from $w' = zx_Rzx$), if the contexts of the form (x_R, x) and (y, y_L) are applied alternatively, however, since their corresponding selectors are same, the contexts need not be applied alternatively and one context can be applied two times to arrive to a contradiction. For $\beta = mdf$ mode, assume that $w' = z\underline{x}z\underline{x}$ or $\underline{x}z\underline{x}z \in L_7$ is derived from $w'' = zz$. Since $w' \in \{a, b\}^*$, and the selector is over $\{a, b\}$ with maximal length, from w' we can derive $\underline{y}zxzx\underline{y} \notin L_7$ or $\underline{y}xzxz\underline{y} \notin L_7$. □

The above result is the converse relation for the Lemma 6. Therefore from the above two lemmas and Lemma 2, we have the following theorem.

Theorem 3. *The families $ICC_{arc}(REG)$ and $ICC_{iemdf}(REG)$ are incomparable with the families $ICC_\beta(REG)$, $\beta \in \{mloc, mdf, emdf\}$, but not disjoint.*

Lemma 8. $ICC_{arc}(REG) - ICC_{iemdf}(REG) \neq \emptyset$.

Proof. Consider the language $L_8 = \{b^n a^m c b^n a^m b^n \mid n, m \geq 0\}$. This language can be generated by $G_{arc} = (\{a, b, c\}, c, (cb^*, \{(b, bb), (a, a)\}), (cb^*a^*, (a, a)))$. Initially, starting with the axiom c, the *arc* grammar generates strings of the form $b^n cb^{2n}, n \geq 1$, using the context (b, bb). As half of the right context is absorbed every time to the selector, the next selector (for the word $b^n cb^2 n$) will be cb^n and now the context (a, a) is applied several times to generate the language L_8. A sample derivation in *arc* mode is given by

$$c \Longrightarrow_{arc} \underline{b}^\downarrow c\underline{b^\downarrow b} \Longrightarrow_{arc} b\underline{b}^\downarrow cb\underline{b^\downarrow} bb \Longrightarrow_{arc} bb\underline{b}^\downarrow cbb\underline{b^\downarrow b}bb \Longrightarrow^*_{arc} b^{n-1}\underline{b}^\downarrow cb^{n-1}\underline{b^\downarrow b}b^{n-1}$$
$$\Longrightarrow_{arc} b^n\underline{a}^\downarrow cb^n\underline{a}^\downarrow b^n \Longrightarrow_{arc} b^n a\underline{a}^\downarrow cb^n a\underline{a}^\downarrow b^n \Longrightarrow^*_{arc} b^n a^m cb^n a^m b^n.$$

However, this language does not belong to $ICC_{iemdf}(REG)$. On contrary, let us assume an *iemdf* grammar generates L_8. Notice that the occurrences of b are pumped equally at three places in the language. In general, no internal contextual grammar can pump more than two occurrences since at every derivation, we adjoin only two contexts. Therefore, the necessary occurrences of b must be pumped before a is pumped, using a context of the form $(b^i, b^{2i}), i \geq 1$, with the associated selector is of the form b^*cb^*. Since the contexts are end-marked, we have either b_L^i or bb_R^{2i}. If bb_R^{2i} is the case, then the occurrences of a cannot be inserted in between bs. When b_L^i is the case, we can only generate $b^ncb^nb^n$ and the occurrences of a cannot be inserted at the correct place on the left of c (a sample derivation is $c \Longrightarrow_{iemdf} {}^{\downarrow}\underline{b_L}c\underline{b^{\downarrow}b} \Longrightarrow_{iemdf} {}^{\downarrow}\underline{b_L}bcb\underline{b^{\downarrow}b}b \Longrightarrow_{iemdf} {}^{\downarrow}\underline{b_L}bbcbb\underline{b^{\downarrow}b}bb \Longrightarrow^* b^ncb^nb^n$). Note that, in *iemdf* mode, the selector should not cover both the adjoined contexts. □

Lemma 9. *$ICC_\alpha(REG) - ICC_{arc}(REG) \neq \emptyset,\ \alpha \in \{iemdf, mdf\}$.*

Proof. Consider the language $L_9 = \{a^nb^ncb^n \mid n \geq 1\}$. This language can be generated by the grammar $G_\alpha = (\{a,b,c\}, abcb, (b^+cb^+, (ab, b_R))),\ \alpha \in \{mdf, iemdf\}$ (for *mdf* grammar, there is no subscript R in the context).

However, $L_9 \notin ICC_{arc}(REG)$. On contrary, let us assume that $L_9 \in ICC_{arc}(REG)$ for an *arc* grammar G_{arc}. As c is a marker in the language, it is easy to see that any context which uses to generate the language will be of the form $(a^ib^i, b^i),\ i \geq 1$, and the associated selector will be of the form $b^{j_1}cb^{j_2}, j_1, j_2 \geq 1$. In this mode, the selector never absorbs the left context. So, there is no change in the left end of the selector in every derivation. In order to generate the strings of the language, at each derivation, the selector should absorb the substring b^i from the left adjoined context a^ib^i. Otherwise, the symbols a and b do not occur in order. This results misplaced occurrences of a and b in the generated string. A contradiction. □

From the above two lemmas and Lemma 2, we have the following result.

Theorem 4. *$ICC_{arc}(REG)$ is incomparable with $ICC_{iemdf}(REG)$, but not disjoint.*

Lemma 10. *$ICC_{mdf}(REG) - ICC_{mloc}(REG) \neq \emptyset$.*

Proof. Consider the language $L_{10} = \{a^ncb^{n+m}da^m \mid n, m \geq 1\}$. This can be generated by $G_{mdf} = (\{a,b,c,d\}, acbbda, (cb^+, (a,b)), (b^+d, (b,a)))$. However this language is not in $ICC_{mloc}(REG)$. Assume that $L_{10} \in ICC_{mloc}(REG)$ for a *mloc* grammar. To generate the language, the grammar will have the contexts of the form (a^i, b^i) and $(b^j, a^j),\ i, j \geq 1$, and their associated selectors will be of the form $a^{k_1}cb^{k_2}$, $b^{k_3}da^{k_4}$, respectively for $k_1, k_2, k_3, k_4 \geq 0$. Consider the word $a^{n'}cb^{n'}bda \in L_{10}$ for a large n' (thus the word is not in the axiom). To reach this word from the axiom, we might have used the context (a^i, b^i) (may be several times) and the selector $a^{k_1}cb^{k_2}$. As we work in *mloc* mode, any further selector must have this selector as a subword. However, from this word, we cannot reach a word $a^{n'}cb^{n'+m'}da^{m'} \in L_{10}$ for a large m'. To reach this word, the selector $b^{k_3}da^{k_4}$ must be used, but it does not have the previously used selector $a^{k_1}cb^{k_2}$ as a substring. A similar argument can be given to the word $a^{n'}cb^{n'+m'}da^{m'}$, if we drive from $acb^{m'}da^{m'}$, for a large m'. □

From Lemma 2, 5 and 10, we have the following result.

Theorem 5. *$ICC_{mdf}(REG)$ is incomparable with $ICC_{mloc}(REG)$, but not disjoint.*

Lemma 11. *$ICC_{mdf}(REG) - ICC_{emdf}(REG) \neq \emptyset$.*

Proof. Consider the language $L_{11} = \{a^n cb^m cb^m ca^n \mid n, m \geq 1\}$. This can be generated by $G_{mdf} = (\{a, b, c\}, acbcbca, (b^+cb^+, (b, b)), (a^*cb^+cb^+ca^*, (a, a)))$. However this language is not in $ICC_{emdf}(REG)$. On contrary, let $L_{11} \in ICC_{emdf}(REG)$ for a *emdf* grammar Assume that first we pump the occurrences of b and then the occurrences of a. As b is equally pumped around the second c, there will be a context of the form $(b^i, b^i), i \geq 1$. As at least one of the context is end-marked, we have either b^i_L or b^i_R. Let the left context be end-marked. Then, if b^i_L is the case, then the next selector should begin with b^i and therefore the left context used in the next derivation should be adjoined to the left of b^i_L. Hence, we cannot pump the occurrences of a on the left of first c, using a context of the form $(a^j, a^j), j \geq 1$. Similarly, if b^i_R is the case, then the next selector should end with b^i and therefore the right context used in the next derivation should be adjoined to the right of b^i. Hence, we cannot pump the occurrences of a on the right of third c, using a context of the form $(a^p, a^p), p \geq 1$. Otherwise, we can produce a word which is not in the language. We can give a similar proof if the right context b^i is end-marked. If we assume that first we pump the occurrences of a and then b, then there should be a context of the form $(a^k, a^k), k \geq 1$, in order to pump the occurrences of a equally at the ends. As one of the contexts is end-marked, we have either a^k_L or a^k_R. We assume that the left context a^k is end-marked (i.e., a^k_L or a^k_R). Then, it is easy to see that we cannot pump the occurrences of b equally around the second c. If the right context a^k is end-marked, we can give a similar reasoning for not pumping the occurrences b equally. □

From Lemma 2, 5 and 11, we have the following result.

Theorem 6. *$ICC_{mdf}(REG)$ is incomparable with $ICC_{emdf}(REG)$, but not disjoint.*

Lemma 12. *$ICC_{emdf}(REG) - ICC_{\beta}(REG) \neq \emptyset$, $\beta \in \{max, mloc, mdf, arc, iemdf\}$.*

Proof. Consider the language $L_{12} = \{a^{2m} ca^{m+n-1} ca^{2n} \mid n, m \geq 1\}$. This can be generated by the grammar $G_{emdf} = (\{a, c\}, \{aacacaa\}, (aa^+ca^+, \{(aa_L, a), (aa, a_L)\}), (acaa^+, (a, aa_R)))$. Intuitively, the first selector aa^+caa^+ is used to increase the necessary occurrences of m and $2m$ of a around the first c and the second selector $acaa^+$ is used to increase the necessary occurrences of n and $2n$ of a around the second c. Whenever, the first selector aa^+ca^+ and the context (aa_L, a) is applied, we can continue further derivations with the same selector aa^+ca^+ itself. If the context (aa, a_L) is applied, in the next derivation the second selector $acaa^+$ must be chosen. Once this selector is chosen, the same selector $acaa^+$ can only be used in the further derivations and choosing the first selector is not possible thereafter. However, this does not affect generating the language L_{12} as the first selector can be used for the required $2m$ and m occurrences of as and then the second selector can be used to generate the required number of n and $2n$ occurrences of as. Note that, due to the maximal condition of the selector, whenever the context (aa_L, a) or (aa, a_L) is applied, the right context a is always adjoined just before the second c and this feature helps to switch over to use the second selector. It is easy to see that $L(G_{emdf}) = L_{12}$.

However, the language L_{12} is not in $ICC_\beta(REG)$ for the above β. Assume that the language $L_{12} \in ICC_\beta(REG)$ for any grammar $G_\beta = (\{a, b, c\}, A, (S_1, C_1), \ldots, (S_r, C_r))$. In order to pump the as around the first c we need a context of the form $(a^{2i}, a^i), i \geq 1$, and the associated selector will be of the form $a^{k_1}ca^{k_2}, k_1, k_2 \geq 0$. Similarly, in order to pump the as around the second c we need a context of the form $(a^j, a^{2j}), j \geq 1$, and the associated selector will be of the form $a^{k_3}ca^{k_4}, k_3, k_4 \geq 0$. Let $\beta = max, mdf$. Assume that $a^{2m'}ca^{m'}a^{n'}ca^{2n'} \in L_{12}$ is obtained by adjoining the context (a^{2i}, a^i) and the selector is used around the second c (i.e., $a^{k_1}ca^{k_2}$); the other case of adjoining the context (a^j, a^{2j}) is similar. Since the right context a^i can be covered by the as in between the two cs (i.e., by the selector $a^{k_3}ca^{k_4}$) and the chosen selector can be locally maximal, we can adjoin the context (a^j, a^{2j}) and derive a word $a^{2m'}\underline{a^j}ca^{m'}\underline{a^{2j}}a^{n'}ca^{2n'} \notin L_{12}$. For $\beta = mloc, arc$ mode, we can generate only one part of the language as we have seen that these variants do not pump symbols across the two markers. The case $\beta = iemdf$ mode is similar to $arc, mloc, mldf$, because the next selector should be inside the adjoined contexts, thus the selector cannot go across the two markers. □

From Lemma 2, 6 and 12, we have the following result.

Theorem 7. *$ICC_{emdf}(REG)$ is incomparable with $ICC_{iemdf}(REG)$, but not disjoint.*

Lemma 13. *$ICC_{mloc}(REG) - ICC_{emdf}(REG) \neq \emptyset$.*

Proof. Consider the language $L_{13} = \{a^n \mid n \geq 1\} \cup \{b^n \mid n \geq 1\} \cup \{a^ncb^n \mid n \geq 1\}$. This can be generated by the $mloc$ grammar $G_{mloc} = (\{a, b, c\}, \{a, aa, b, bb, acb\}, (aa, (\lambda, a)), (bb, (\lambda, b)), (acb, (a, b)))$. However, this language cannot be generated by an $emdf$ grammar. Assume that $L_{13} \in ICC_{emdf}(REG)$ for any grammar $G_{emdf} = (\{a, b, c\}, A, (S_1, C_1), \ldots, (S_r, C_r))$. In order to generate the strings a^n and b^n, we need contexts of the form $(a_E^{i_1}, a_E^{i_2}),\ i_1 + i_2 \geq 1$, and $(b_E^{j_1}, b_E^{j_2}),\ j_1 + j_2 \geq 1$, with their associated selectors of the form $a^i, i \geq 1$, and $b^j, j \geq 1$. Also, in order to generate the strings a^ncb^n, we need a context of the form $(a_E^k, b_E^k), k \geq 1$, with the associated selector of the form $a^{k_1}cb^{k_2},\ k_1, k_2 \geq 1$. The suffix E denotes the (right or the left) end-marker. Consider a word $a^{n'}cb^{n'} \in L_{12}$ for a large n'. Then, to reach this word, we should have used the context (a_E^k, b_E^k). In $emdf$ mode, at each derivation, the selector should cover and start/end with one of the adjoined contexts. Therefore, the next selector should start or end with the context a^k (the other case for the context b^k is similar). Such a context can be covered by a selector $a^{m'}$, thus we can apply the context (a^{i_1}, a^{i_2}) to $a^{n'}cb^{n'}$, resulting a word $a^{n'+i_1+i_2}cb^{n'} \notin L_{13}$. A contradiction. □

From the above two lemmas and Lemma 2, we have the following theorem.

Theorem 8. *$ICC_{mloc}(REG)$ is incomparable with $ICC_{emdf}(REG)$, but not disjoint.*

4 Conclusion

In this paper, we have considered the generative power of various classes of internal contextual grammars where the restrictions are considered in the derivations, namely, $max, mloc, mdf, arc, emdf, iemdf$. We conjecture that Lemma 2 can be strengthened as the class of regular languages is in the family of languages $ICC_{\alpha}(REG)$ for the variants discussed in this paper.

In the Chomsky hierarchy of languages, when the restrictions are increased in the form of production rules (from unrestricted to context sensitive (i.e. context dependent), from context dependent to context-free, from context-free to regular), the generative power of the class of grammars is decreased. On the other hand, in regulated rewriting, when the rules are context-free (for instance, matrix grammars, programmed grammars, periodically time varying grammars and grammars with regular control), putting restrictions in the manner of applying the rules, the generative power of the grammars is increased (but for type-3 rules of regulated rewriting, the generative power is unaltered) [3],[22]. Therefore, it will be a nice result in the field of formal languages to show that there are families of languages whose grammars are obtained by imposing more restrictions on the manner of applying the rules, but the generative power of the grammars is neither increased nor decreased; they are incomparable. In this paper, we have identified the families of languages in the domain of contextual grammars which possess this interesting property. Also, we showed that there are languages which are common to all these families of languages. Hence these families are not disjoint.

Thus, we have found that there is a class of languages obtained by putting restrictions in the derivation of the same basic class of grammars (internal contextual grammars) whose behaviour is different from the existing class of grammars in formal languages theory. How these restrictions play a role in natural language processing is an interesting problem which could be explored in future. A study of descriptional complexity measures of the internal contextual grammars under these restrictions can also be explored. We refer to [10], [13], [14] for recent works where descriptional complexity measures of internal contextual grammars and ambiguity of contextual languages were considered.

Acknowledgments. The authors thank the three anonymous referees for their insightful remarks and comments which helped to improve the presentation of the paper. The first author's work was partially carried out during the tenure of an ERCIM "Alain Bensoussan" Fellowship Programme. Also, he mentions his special thanks to Dr. Eric Villemonte de la Clergerie, INRIA, for his support and encouragement.

References

1. Boullier, P.: Range concatenation grammars. In: Proceedings of Sixth International Workshop on Parsing Technologies (IWPT 2000), pp. 53–64 (2000)

2. Boullier, P.: From contextual grammars to range concatenation grammars. Electronic Notes in Theoretical Computer Science 53, 41–52 (2001)
3. Dassow, J., Păun, G.: Regulated Rewriting in Formal Language Theory. EATCS monographs on TCS. Springer, Heidelberg (1989)
4. Ehrenfeucht, A., Ilie, L., Păun, G., Rozenberg, G., Salomaa, A.: On the generative capacity of certain classes of contextual grammars. In: Mathematical Ling. and Related Topics, pp. 105–118. The Publ. House of the Romanian Academy, Bucharest (1995)
5. Joshi, A.K.: How much context-sensitivity is required to provide structural descriptions: Tree adjoining grammars. In: David, D., Lauri, K., Arnold, Z. (eds.) Natural Language Processing: Psycholinguistic, Computational, and Theoretical Perspectives, pp. 206–250. Cambridge University Press, New York (1985)
6. Ilie, L.: A non-seminlinear language generated by an internal contextual grammar with finite selection. Ann. Univ. Bucharest Math. Inform. Series 45, 63–70 (1996)
7. Ilie, L.: On computational complexity of contextual languages. Theo. Comp. Science 183(1), 33–44 (1997)
8. Ilie, L.: Some recent results in contextual grammars. Bull. EATCS 62, 172–194 (1997)
9. Lakshmanan, K.: New Classes of Contextual Grammars for Mildly Context Sensitive Formalisms. In: Terikhovsky, O.N., Burton, W.N. (eds.) New Topics in Theoretical Computer Science, pp. 1–25. Nova Publishers, USA (2008)
10. Lakshmanan, K., Anand, M., Krithivasan, K.: On the trade-off between ambiguity and measures in internal contextual grammars. In: Proceedings of 10th International Workshop on Descriptional Complexity of Formal Systems, pp. 216–223 (2008)
11. Lakshmanan, K., Krishna, S.N., Rama, R., Martin-Vide, C.: Internal contextual grammars for mildly context sensitive languages. Research on Language and Computation 5(2), 181–197 (2007)
12. Lakshmanan, K.: End-marked maximal depth-first contextual grammars. In: Ibarra, O.H., Dang, Z. (eds.) DLT 2006. LNCS, vol. 4036, pp. 339–350. Springer, Heidelberg (2006)
13. Lakshmanan, K.: A note on Ambiguity of Internal Contextual Grammars. Theo. Comp. Science 369, 436–441 (2006)
14. Lakshmanan, K.: Incompatible measures of internal contextual grammars. In: Proceedings of DCFS 2005, pp. 253–260 (2005)
15. Marcus, S.: Contextual grammars. Rev. Roum. Pures. Appl. 14, 1525–1534 (1969)
16. Marcus, S., Martin-Vide, C., Păun, Gh.: Contextual grammars as generative models of natural languages. Computational Linguistics 24(2), 245–274 (1998)
17. Marcus, S., Martin-Vide, C., Păun, Gh.: On internal contextual grammars with maximal use of selectors. In: Proc. 8th Conf. Automata & Formal Lang., vol. 54, pp. 933–947. Salgotarjan. Publ. Math., Debrecen (1999)
18. Martin-Vide, C., Miquel-Verges, J., Păun, Gh.: Contextual grammars with depth-first derivation. In: Tenth Twente Workshop on Language Tech.; Algebraic Methods in Language Processing, Twente, pp. 225–233 (1995)
19. de Groote, P.: Towards abstract categorial grammars. In: ACL 2009. ACL Press (2001), `http://www.aclweb.org/anthology/P01-1-33`
20. Păun, Gh., Nguyen, X.M.: On the inner contextual grammars. Rev. Roum. Pures. Appl. 25, 641–651 (1980)
21. Păun, Gh.: Marcus Contextual Grammars. Kluwer Academic Publishers, Dordrecht (1997)
22. Salomaa, A.: Formal Languages. Academic Press, London (1973)