




Centrum Wiskunde & Informatica

View metadata, citation and similar papers at core.ac.uk

brought to you by  CORE

provided by CWI's Instituut

REPORTRAPPORT

SEN

Software Engineering



Software ENgineering

Semi-bracketed contextual grammars

L. Kuppusamy

REPORT SEN-R0808 DECEMBER 2008

Centrum Wiskunde & Informatica (CWI) is the national research institute for Mathematics and Computer Science. It is sponsored by the Netherlands Organisation for Scientific Research (NWO). CWI is a founding member of ERCIM, the European Research Consortium for Informatics and Mathematics.

CWI's research has a theme-oriented structure and is grouped into four clusters. Listed below are the names of the clusters and in parentheses their acronyms.

Probability, Networks and Algorithms (PNA)

Software Engineering (SEN)

Modelling, Analysis and Simulation (MAS)

Information Systems (INS)

Copyright © 2008, Centrum Wiskunde & Informatica
P.O. Box 94079, 1090 GB Amsterdam (NL)
Kruislaan 413, 1098 SJ Amsterdam (NL)
Telephone +31 20 592 9333
Telefax +31 20 592 4199

ISSN 1386-369X

Semi-bracketed contextual grammars

ABSTRACT

Bracketed and fully bracketed contextual grammars were introduced to bring the concept of a tree structure to the strings by associating a pair of parentheses to the adjoined contexts in the derivation. In this paper, we show that these grammars fail to generate all the basic non-context-free languages, thus cannot be a syntactical model for natural languages. To overcome this failure, we introduce a new class of fully bracketed contextual grammars, called the semi-bracketed contextual grammars, where the selectors can also be non-minimally Dyck covered language. We see that the tree structure to the derived strings is still preserved in this variant. when this new grammar is combined with the maximality feature, the generative power of these grammars is increased to the extend of covering the family of context-free languages and some basic non-context-free languages, thus possessing many properties of the so called 'MCS formalism'.

2000 Mathematics Subject Classification: 68Q45

1998 ACM Computing Classification System: F.4.2

Keywords and Phrases: contextual grammars, MCS formalism, derivation tree structure

Note: The work was carried out during the tenure of an ERCIM "Alain Bensoussan" Fellowship Programme.

Semi-Bracketed Contextual Grammars

Lakshmanan Kuppusamy*
Centrum Wiskunde en Informatica
413 Kruislaan, 1098 SJ
Amsterdam, The Netherlands.
L.Kuppusamy@cwi.nl

Abstract

Bracketed and fully bracketed contextual grammars were introduced [16] to bring the concept of a tree structure to the strings by associating a pair of parentheses to the adjoined contexts in the derivation. In this paper, we show that these grammars fail to generate all the basic non-context-free languages, thus cannot be a syntactical model for natural languages. To overcome this failure, we introduce a new class of fully bracketed contextual grammars, called the semi-bracketed contextual grammars, where the selectors can also be non-minimally Dyck covered language. We see that the tree structure to the derived strings is still preserved in this variant. when this new grammar is combined with the *maximality* feature, the generative power of these grammars is increased to the extend of covering the family of context-free languages and some basic non-context-free languages, thus possessing many properties of the so called ‘MCS formalism’.

Keywords: contextual grammars, MCS formalisms, derivation tree structure.

1 Introduction

Contextual grammars were introduced by S. Marcus in 1969. They produce languages starting from a finite set of *axioms* and adjoining *contexts*, iteratively, according to a *selector* present in the current sentential form. As introduced in [13], if adjoining the contexts is done at the ends of the strings, the grammar is called *external*. *Internal* contextual grammars were introduced by Păun and Nguyen in 1980 [19], where the contexts are adjoined to the selector strings appearing as substrings of the string. Later on, many variants of contextual grammars were introduced and we refer to [5],[9],[10],[20] for some of them.

One of the important problems in the area of formal language theory and natural language processing is to obtain certain classes of languages that provide an appropriate description for natural languages. In fact, the classes of languages searched for should have the so called ‘mildly context sensitive’ (MCS) properties which are defined as follows:

1. The class of languages contains all context-free languages
2. The class of languages contains the following three basic non-context-free languages:
 - *multiple agreements*: $L_1 = \{a^n b^n c^n | n \geq 1\}$,
 - *crossed dependencies*: $L_2 = \{a^n b^m c^n d^m | n, m \geq 1\}$, and
 - *marked duplication*: $L_3 = \{w c w | w \in \{a, b\}^*\}$.

*The work was carried out during the tenure of an ERCIM "Alain Bensoussan" Fellowship Programme.

3. All the languages in the class are *parsable in polynomial time*.
4. All the languages in the class have the *bounded growth property*. That is, for any infinite language L , there is a constant k_L such that, for any $n \geq 1$, if L contains words of length n , then it also contains words of some length between $n + 1$ and $n + k_L$. In other words, informally, there should not be arbitrarily large gaps between two consecutive words present in the language.

A class of languages possesses the MCS properties characterize the MCS family of languages and the corresponding class of grammars forms the MCS formalism. For more details on MCS formalisms, we refer to [7], [17]. One such example for MCS formalisms is Tree Adjoining Grammar (TAG) [8]. We make a clear note here that so far no unanimously agreed definition for mildly context sensitive formalism is available. For example, in [5], the condition of context-free inclusion was not present but *semilinear* condition was present instead of the last condition. However, in this paper, we stick to the initial conditions specified above for mildly context sensitive formalism.

Even though contextual grammars were introduced to give an appropriate model description for natural languages [12], the basic class, internal contextual languages itself fails to contain non-context-free constructions [3], [4]. Further, the membership problem for the above families of languages still remains open [6], or is solvable only with exponential time [2].

In the last decade or so, some attempts have been made to introduce some variants of contextual grammars by restricting the selector chosen in the derivation to obtain certain specific classes of contextual languages and therefore several new classes of grammars were introduced, for instance, *global maximal* and *local maximal* [14]. In these modes, at each derivation step, a selector having maximal length (either with respect to all selectors or with respect to the local selector) is chosen for the next derivation. Though these variants generate non-context-free languages, they still allow non-semilinear languages [17] and they can be parsed with exponential time complexity by transforming the given grammar into an equivalent *dynamic range concatenation grammar* [1]).

Unlike as context-free grammars, no derivation tree exists for contextual grammars. In order to define a derivation tree for a derivation in internal contextual grammars, bracketed and fully bracketed contextual grammars were introduced in [16]. The selectors for these grammars are based on the notion of minimally Dyck covered language, which is a restricted version of the well known Dyck covered language. In these grammars, a notion of structure to the strings was introduced, by associating a pair of parentheses (brackets) to the contexts inserted at each derivation step. The generative power of these grammars has been discussed in [15]. However, the relevance of these grammars to MCS formalisms has not been explored in detail so far.

In this paper, we first investigate the power of fully bracketed contextual grammars for the above mentioned natural language constructions. In [16], it was shown that fully bracketed contextual grammars with regular selectors cannot generate the crossed dependency language. Here, we prove that these grammars with regular selectors cannot cover the other two non-context-free constructions. To overcome this failure, we introduce a new variant called semi-bracketed contextual grammars obtained by relaxing the condition in fully bracketed contextual grammars that the selectors need be minimally Dyck covered string. We see that the structure to the strings is still maintained in this new variant. When the maximality condition is incorporated with this variant, we show that the basic non-context-free languages are covered with regular selectors. We also show an important result that the family of context-free languages is contained in this variant, a rare result in the domain of contextual languages.

2 Preliminaries

In this section, we introduce the notion of formal languages and contextual grammars which are used in the paper. A finite non-empty set V is called an *alphabet*. We denote by V^* the free monoid generated by V , by λ its identity or the *empty string*, and by V^+ the set $V^* - \{\lambda\}$. The elements of V^* are called *words*. For any word $x \in V^*$, we denote by $|x|$ the *length of x* . $|x|_a$ is the number of occurrences of the symbol a in the word x . The families of *finite*, *regular*, *context-free* and *context-sensitive* languages are denoted by *FIN*, *REG*, *CF* and *CS* respectively. For more details on formal language theory, we refer to [21].

Now we shall present some basic definitions of contextual grammars. An *internal contextual grammar* is a construct

$$G = (V, A, (S_1, C_1), \dots, (S_m, C_m)), \quad m \geq 1, \text{ where}$$

- V is an alphabet,
- $A \subseteq V^*$ is a finite set called the set of *axioms*,
- $S_i \subseteq V^*$, $1 \leq i \leq m$, are the sets of *selectors* (not necessarily of finite type),
- $C_i \subseteq V^* \times V^*$, C_i finite, $1 \leq i \leq m$, are the sets of *contexts*.

The *modular presentation* [18] of a contextual grammar is given as $G = (V, A, P)$ where V, A are as defined above and P is the finite set of selector-context rules of the form $(S_1, C_1), \dots, (S_m, C_m)$. The usual derivation in the *internal mode* (denoted by *in*) is defined as

$$x \Longrightarrow_{in} y \text{ iff } x = x_1x_2x_3, \quad y = x_1ux_2vx_3, \text{ for } x_1, x_2, x_3 \in V^*, \\ x_2 \in S_i, (u, v) \in C_i, \text{ for some } 1 \leq i \leq m.$$

Consider a contextual grammar $G = (V, A, (S_1, C_1), \dots, (S_m, C_m))$. The *maximal mode* of the grammar is defined in the following way [14].

$$x \Longrightarrow_M y \text{ iff } x = x_1x_2x_3, \quad y = x_1ux_2vx_3, \text{ for } x_2 \in S_i, (u_i, v_i) \in C_i, \quad 1 \leq i \leq m, \\ \text{and there are } \mathbf{no} \ x'_1, x'_2, x'_3 \in V^*, \text{ such that } x = x'_1x'_2x'_3, \quad x'_2 \in S_i, \text{ and } x'_2 \text{ contains } x_2.$$

That is, in this maximal mode, the chosen selector x_2 for the next derivation should not be contained (in substring sense) in a longer selector x'_2 , where both $x_2, x'_2 \in S_i$ for some i . The language generated by a contextual grammar G in internal (respectively maximal mode) is given as $L_\alpha(G) = \{x \in V^* \mid z \Longrightarrow_\alpha^* x, z \in A\}$, where \Longrightarrow_α^* is the reflexive transitive closure of the relation \Longrightarrow_α and $\alpha \in \{in, M\}$.

Let us consider the brackets $[,]$ and denote the set $\{[,]\}$ by B . The Dyck language over B is denoted by D_B and it is the language generated by the context-free grammar $G = (\{S\}, B, S, \{S \rightarrow SS, S \rightarrow [S], S \rightarrow \lambda\})$. Given the two disjoint sets V and B , we can define the projection mapping pr_V, pr_B , from $(V \cup B)^*$ to V^*, B^* , respectively as follows:

$$pr_\beta(a) = \begin{cases} a, & \text{for } a \in \beta \\ \lambda, & \text{for } a \notin \beta, \end{cases}$$

where $\beta \in \{V, B\}$. A string $x \in (V \cup B)^*$ is said to be a *Dyck covered string* if $x \Longrightarrow^* \lambda$, by reduction rules of the form $[w] \rightarrow \lambda$, for $w \in V^*$. For instance, $x_1 = [[a]a[a]]$, $x_2 = [a][a]$, and $x_3 = [[a][a]]$ are Dyck covered strings. Clearly, if $x \in (V \cup B)^*$ is a Dyck covered string, then $pr_B(x) \in D_B$. A Dyck covered string $x \in (V \cup B)^*$ is said to be *minimally Dyck covered string* if the following conditions are hold:

1. if $x = x_1]x_2[x_3$ with $x_1, x_3 \in (V \cup B)^*$ and $x_2 \in V^*$, then $x_2 = \lambda$.
2. The reduction rule $[] \rightarrow \lambda$ is not used when reducing x to λ .

Condition 1 refutes the string x_1 above, condition 2 refutes the string x_3 , hence these strings are not minimally Dyck covered; the string x_2 is of this type. We denote the language of all minimally Dyck covered strings over the alphabet V by $MDC(V)$.

For every 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 all the leaves down;
- scan x from the left to right and grow $\tau(x)$ according to the following two rules:
 - for each *maximal* substring $[w$ of x , with $w \in V^*$ (since w is maximal, after w we find either $[$ or $]$), we draw a new edge, starting at the current point of the partially constructed $\tau(x)$, marked with w on its left side, and placed to the right hand of the currently constructed tree;
 - for each maximal $w]$, $w \in V^*$, not scanned yet (hence, either we find $]$ before w , 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.

In figure 1, a derivation tree $\tau(x)$ is drawn for the word $x = [a[ab][ab[ab[c]b]b]a][a]$.

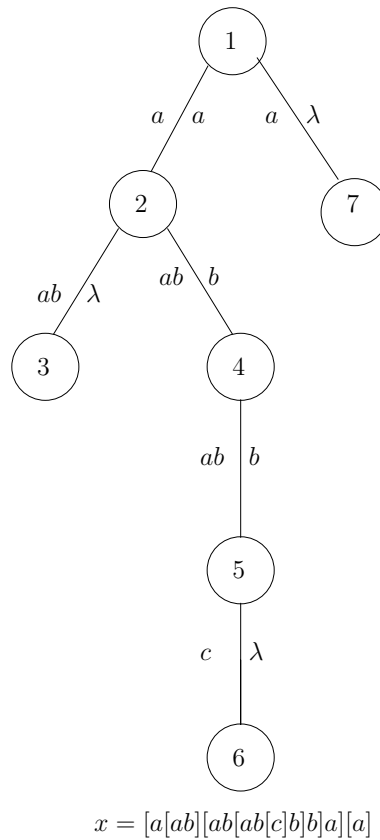


Figure 1: derivation tree structure $\tau(x)$

Now, let us recall the definitions of bracketed and fully bracketed internal contextual grammars which originated from the above concepts in order to give a tree structure to the derived strings. A *bracketed contextual grammar* is a tuple $G = (V, A, (S_1, C_1), \dots, (S_m, C_m))$, $m \geq 1$ where V is an alphabet, A is a finite subset of $MDC(V)$, called axioms, $S_i \subseteq V^*$, and C_i are finite subsets of $V^* \times V^* - \{(\lambda, \lambda)\}$ for all $1 \leq i \leq m$. The derivation relation (in internal contextual mode) is defined as follows: for $x, y \in (V \cup B)^*$, we write $x \Longrightarrow_G y$, if and only if $x = x_1x_2x_3$, $y = x_1[ux_2v]x_3$, $x_1, x_3 \in (V \cup B)^*$, $x_2 \in MDC(V)$ and $pr_V(x_2) \in S_i$, $(u, v) \in C_i$, for some $1 \leq i \leq m$.

A *fully bracketed contextual grammar* (in short, FBIC grammar) is similar to a bracketed contextual grammar, except that the selectors are in $MDC(V)$ instead of $S_i \subseteq V^*$, and no projection is applied to the chosen selector. It is proved in [16] that if $x \Longrightarrow_G y$ is a derivation step in a bracketed or fully bracketed contextual grammar, then $y \in MDC(V)$ whenever $x \in MDC(V)$. Note that, since the axioms of these grammars are in $MDC(V)$, any string derived under these modes is also in $MDC(V)$, thus each derived string will have a tree structure, as mentioned above.

The *string language* generated by a bracketed or fully bracketed contextual grammar $G = (V, A, (S_1, C_1), \dots, (S_m, C_m))$, $m \geq 1$ is defined as

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

where \Longrightarrow_G^* is the one discussed already. Moreover, we can associate to G , the bracketed language $BL(G)$ defined by

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

Unless otherwise specified for each grammar, we assume the projection of V as the one discussed above. The selectors of a bracketed contextual grammar are languages over V , hence their type is clear. But, in the case of FBIC grammars, the brackets are important when discussing the type of the selectors, thus leading to a confusion in the type. To avoid this, the type of projection $pr_V(S_i)$ is considered for the selectors of FBIC grammars. That is, the brackets are ignored while considering the selector type for FBIC grammars. When the selectors are of regular languages for grammars, we represent them as a form of regular expressions itself instead of language of regular expressions. The family of languages generated by internal, maximal, bracketed, fully bracketed contextual grammars with F selectors (or choice) is denoted by $ICC(F)$, $MIC(F)$, $BIC(F)$, $FBIC(F)$, respectively. For more technical details on contextual grammars, we refer to the monograph [18].

3 Power of FBIC for Non-Context-free Languages

In this section, we analyze the power of FBIC grammars towards generating the non-context-free languages and we show that these grammars fail to generate the non-context-free languages.

Lemma 3.1 $L_1 = \{a^n b^m c^n d^m \mid n, m \geq 1\} \notin FBIC(REG)$

Proof. See [16].

Lemma 3.2 $L_2 = \{a^n b^n c^n \mid n \geq 1\} \notin FBIC(REG)$

Proof. Assume that the language L_2 can be generated by a fully bracketed contextual grammar $G_2 = (\{a, b, c\}, A, (S_1, C_1), \dots, (S_m, C_m))$ with regular selector. Since the occurrences of a, b and c are pumped evenly by the contexts for the strings of the language, there must exist a context of the form $(a^i b^i, c^i)$, $i \geq 1$ or $(a^j, b^j c^j)$, $j \geq 1$ or $(a^s b^{r_1}, b^{r_2} c^s)$, $s \geq 1$, $r_1 + r_2 =$

s. Assume that $x \Longrightarrow_{G_2} y$ is a derivation in $L(G_2)$, where $x = x_1x_2x_3$ and $x_1x_2x_3$ is in $MDC(V)$. If the context (a^ib^i, c^i) is used in a derivation, that results a string of the form $x_1[\underline{a^ib^i}x_2\underline{c^i}]x_3 \in L_2$. If the context (a^j, b^jc^j) is used in a derivation, that results a string of the form $x_1[\underline{a^j}x_2\underline{b^jc^j}]x_3 \in L_2$. If the context $(a^sb^{r_1}, b^{r_2}c^s)$ is used in a derivation, that results a string of the form $x_1[\underline{a^sb^{r_1}}x_2\underline{b^{r_2}c^s}]x_3$. In the above cases, x_2 is the selector and therefore in $MDC(V)$ and the underlined symbols are the contexts introduced in the current derivation $x \Longrightarrow_{G_2} y$. However, no further derivation after $x \Longrightarrow_{G_2} y$ is possible since no context can further be adjoined. To adjoin any context further in the next derivation step (there are three types of contexts available for the grammar which are discussed above), the selector must be in $MDC(V)$ and it should either start with $[b$, in order to adjoin (a^ib^i, c^i) at the right place or the selector must end with $b]$, in order to adjoin (a^j, b^jc^j) at the right place or the selector must start with $[b$ and end with $b]$, in order to adjoin the context $(a^sb^{r_1}, b^{r_2}c^s)$ at the right place. Such a selector with $MDC(V)$ is not possible in the derived strings: For $x_1[\underline{a^ib^i}x_2\underline{c^i}]x_3$, there is no substring starting $[b$ after a occurs in this word; For $x_1[\underline{a^j}x_2\underline{b^jc^j}]x_3$, there is no substring ending with $b]$ before c occurs in this string; For $x_1[\underline{a^sb^{r_1}}x_2\underline{b^{r_2}c^s}]x_3$, there is no substring that neither starts with $[b$ nor ends with $b]$ in the string. It follows that the derivation is terminated and thus the language generated by the fully bracketed contextual grammar is finite. This is a contradiction to L_2 is infinite. \square

Lemma 3.3 $L_3 = \{xcx \mid x \in \{a, b\}^*\} \notin FBIC(REG)$

Proof. Assume that the language $L_3 = L(G_3)$ for some fully bracketed contextual grammar $G_3 = (\{a, b, c\}, A, (S_1, C_1), \dots, (S_m, C_m))$ with regular selector. In order to generate strings of the form xcx , we have to use a context of the form $(w^i, w^i), i \geq 1, w \in \{a, b\}$, thus producing a string in $MDC(V)$ of the form $x_1[\underline{w^ix_2w^i}]x_3$. It is obvious that any selector for the language L_3 must be of the form cx or xc where $x \in \{a, b\}^*$ and x is of maximal length. Since $x_2 \in MDC(V)$, x_2 is of the form as $x_2 = [cx]$ or $x_2 = [xc]$. To adjoin a context further to the obtained word $x_1[\underline{w^ix_2w^i}]x_3$, the selector must be of the form $x_1[\underline{w^ix_2}$ or $x_2\underline{w^i}]x_3$. Such a selector is not minimally Dyck covered string since the corresponding pair of right or left parenthesis is not present in $x_1[\underline{w^ix_2}$ or $x_2\underline{w^i}]x_3$, respectively. Therefore no context can be adjoined further and this follows that the generated language is finite, a contradiction. \square

From the above proofs, we can see that the problem is due to the condition that the selectors are minimally Dyck covered languages. In the next section, we relax this condition to the selector. Interestingly, the derived strings are yet in $MDC(V)$ and thus the tree structure to the strings is maintained.

4 Semi-Bracketed Contextual Grammars

In this section, first we define a new variant of fully bracketed contextual grammars, namely semi-bracketed contextual grammars. A *semi-bracketed contextual grammar* (in short, SBIC grammar) is a construct $G = (V, A, (S_1, C_1), \dots, (S_m, C_m)), m \geq 1$ where V is an alphabet, $A \in MDC(V)$ is a finite set of axiom, $S_i \subseteq [(V \cup B)^* \cup (V \cup B)^*]$ and C_i are finite subsets of $V^* \times V^* - \{(\lambda, \lambda)\}$ for $1 \leq i \leq m$, with the condition that whenever C_i contains a context $(u, \lambda), u \in V^+$ for some i , then the corresponding selector is of the form $S_i \in [(V \cup B)^*$ and whenever C_i contains a context $(\lambda, v), v \in V^+$ for some i , then the corresponding selector is of the form $S_i \in (V \cup B)^*]$. Note that, when the context is not one-sided (one sided means either $u = \lambda$ or $v = \lambda$ in (u, v)), the corresponding selector may be of any type; may start or end with a bracket. The derivation relation is defined as follows. For $x, y \in (V \cup B)^*$, we write $x \Longrightarrow_G y$ if and only if $x = x_1x_2x_3, y = x_1[ux_2v]x_3$, where $x_1, x_3 \in (V \cup B)^*, x_2 \in S_i, (u, v) \in C_i$, for some $1 \leq i \leq m$.

Next, we shall show that each derivation step of the above introduced grammar preserves the minimally Dyck covered string.

Lemma 4.1 *If $x \Longrightarrow_G y$ in a semi-bracketed contextual grammar $G = (V, A, (S_1, C_1), \dots, (S_m, C_m))$ and $x \in MDC(V)$, then $y \in MDC(V)$.*

Proof. Assume that $x = x_1x_2x_3$, $y = x_1[ux_2v]x_3$, for $(u, v) \in C_i$ such that $x_1x_2x_3 \in MDC(V)$ and $x_2 \in S_i$ for some i . As per the definition of the grammar, the selectors can be of three types. First, let us assume that $S_i \subseteq [(V \cup B)^*]$. Then, x can be rewritten as $x = x_1[x'_2x'_3]x''_3$ with $x_1, x'_3, x''_3 \in (V \cup B)^*$, $[x'_2] \in S_i$, and $x_1x''_3 \in MDC(V)$. Similarly, y can be rewritten as $y = x_1[u[x'_2v]x'_3]x''_3$. By applying the *MDC* reduction rule to y , we get $[x'_2v] \rightarrow \lambda$, $[ux'_3] \rightarrow \lambda$, and $x_1x''_3 \rightarrow \lambda$, thus $y \in MDC(V)$. Note that, the result also holds true for contexts of the form (u, λ) , $u \in V^+$ whenever x'_2 contains at least one symbol from V . Also, the reduction does not necessarily take place in one step. If $S_i \subseteq (V \cup B)^*$, then, x can be rewritten as $x = x'_1[x''_1x'_2]x_3$ with $x'_1, x''_1, x_3 \in (V \cup B)^*$, $[x'_2] \in S_i$, and $x'_1x_3 \in MDC(V)$. Similarly, y can be rewritten as $y = x'_1[x''_1[u[x'_2v]]x_3$. By applying the *MDC* reduction rule to y , we get $[ux'_2] \rightarrow \lambda$, $[x''_1v] \rightarrow \lambda$, and $x'_1x_3 \rightarrow \lambda$, thus $y \in MDC(V)$. The result also holds true for contexts of the form (λ, v) , $v \in V^+$, whenever x'_2 contains at least one symbol from V . Thirdly, when the selectors are of minimally Dyck covered string type, it is obvious that the derived string is also in *MDC(V)* (since the case reduces to FBIC grammar, for which the result is already proved in [16]). \square

When the maximality condition (i.e., choosing the selector of maximal length) is included with these semi-bracketed contextual grammars, the grammar is said to be a *maximal semi-bracketed* contextual grammar and is denoted by *MSBIC* grammar. The string language generated by a semi-bracketed or by a maximal semi-bracketed contextual grammar G is defined by $L(G) = \{pr_V(w) \mid z \Longrightarrow_G^* w, w \in (V \cup B)^*, z \in A\}$, where \Longrightarrow_G^* is the reflexive transitive closure of the relation \Longrightarrow_G . When G is obvious, it may be omitted in the derivation. Sometimes in a derivation, the projection of a derived word w , $w \in (V \cup B)^*$ with respect to V is denoted as $\Longrightarrow_{pr(V)} w$. It is easy to see that when $\lambda \notin L(G)$, w can be rewritten as $w \in (V \cup B)^*a^+$, where $a \in V$. The family of languages generated by maximal semi-bracketed and semi-bracketed internal contextual grammars with F selector (or choice) is denoted by *MSBIC(F)*, *SBIC(F)*, respectively. The selector type of SBIC, MBHIC grammars are defined by the way of FBIC grammars. In this paper, we consider the selector types as *FIN* and *REG* only.

5 Covering Non-Context-free Languages by MSBIC Grammars

In this section, we first prove that all the three basic non-context-free languages can be realized by this new variant when the chosen selector is of maximal length.

Lemma 5.1 $L_1 = \{a^n b^m c^n d^m \mid n, m \geq 1\} \in MSBIC(REG)$

Proof. Consider a grammar $G_1 = (\{a, b, c, d\}, \{[a[bc]d]\}, \{([a([b]^*[bc], (a, c)), ([bc](c)^*d], (b, d))\})$. Any derivation in the grammar G_1 is given as

$$[a^\downarrow[bc]d]^\downarrow \Longrightarrow [a^\downarrow[b[bc]d]^\downarrow d] \Longrightarrow^\downarrow [a[b[b[bc]^\downarrow d]^\downarrow d]d] \Longrightarrow [a^\downarrow[a[b[b[bc]^\downarrow c]^\downarrow d]d]d] \Longrightarrow_{pr(V)}^* a^n b^m c^n d^m.$$

The substring between the two down arrows indicate the selector used in the next derivation step and the adjoined contexts in a derivation step are marked with underline. It is easy to see that $L(G_1) = L_1$, thus $L_1 \in MSBIC(REG)$. \square

Lemma 5.2 $L_2 = \{a^n b^n c^n \mid n \geq 1\} \in MSBIC(REG)$.

Proof. Consider the grammar $G_2 = (\{a, b, c\}, \{[a[bc]]\}, \{(b^*[bc], (ab, c))\})$. Any derivation of the grammar is given as

$$[a^\downarrow[bc]^\downarrow] \Longrightarrow [a[a^\downarrow b[bc]^\downarrow c]^\downarrow] \Longrightarrow [a[a[a^\downarrow bb[bc]^\downarrow c]^\downarrow c]^\downarrow] \Longrightarrow [a[a[a[a^\downarrow bbb[bc]^\downarrow c]^\downarrow c]^\downarrow c]^\downarrow] \Longrightarrow_{pr(V)}^* a^n b^n c^n.$$

It is easy to see that $L(G_2) = L_2$, thus $L_2 \in MSBIC(REG)$. \square

Lemma 5.3 $L_3 = \{xcx \mid x \in \{a, b\}^*\} \in MSBIC(REG)$

Proof. Consider the grammar $G_3 = (\{a, b, c\}, \{[c]\}, \{([c]({a, b})^*), \{(a, a), (b, b)\})\}$. Any derivation in the grammar is given as

$$[c]^\downarrow \Longrightarrow [w_1^\downarrow [c] w_1]^\downarrow \Longrightarrow [w_1 [w_2^\downarrow [c] w_1] w_2]^\downarrow \Longrightarrow [w_1 [w_2 [w_3^\downarrow [c] w_1] w_2] w_3]^\downarrow \Longrightarrow_{pr(V)}^* xcx,$$

where $w_i = a, b, 1 \leq i \leq n$. It is clear that $L(G_3) = L_3$, thus $L_3 \in MSBIC(REG)$. \square

The following lemma proves that MSBIC grammars with regular choice can even cover the non-marked duplication language, an interesting feature not available to all contextual grammars which are shown to have MCS formalism.

Lemma 5.4 $L'_3 = \{xx \mid x \in \{a, b\}^*\} \in MSBIC(REG)$

Proof. Consider the grammar $G'_3 = (\{a, b\}, \{\lambda, [[a]a], [[b]b]\}, \{(\{w'[w] \mid w' \in \{V, B\}^*, w \in \{a, b\}, \{(a, a), (b, b)\})\})\}$.

Any derivation in the grammar is given as

$$[[w]^\downarrow w] \Longrightarrow [w_1^\downarrow [[w] w_1] w] \Longrightarrow [w_2 [w_1 [w]^\downarrow w_2] w_1] w] \Longrightarrow [w_3 [w_2 [w_1 [w]^\downarrow w_3] w_2] w_1] w] \Longrightarrow_{pr_V}^* xx$$

where $w, w_i = a, b, 1 \leq i \leq n$. Note that $[w]$ is a unique structure in the string and also center in the string, thus it served as a marker of the string. It is easy to see that $L(G'_3) = L'_3$, thus $L'_3 \in MSBIC(REG)$. \square

We next prove an important result by taking advantage of the projection and maximality features in maximal semi-bracketed contextual grammars.

Theorem 5.5 $CF \subseteq MSBIC(REG)$

Proof. Let $G = (N, T, S, P)$ be a context-free grammar in Chomsky normal form, where N is the set of non-terminals (or variables), T is the set of terminals, $S \in N$, the start variable and P is the set of production rules of the form $A \rightarrow CD$, $A \rightarrow a$, $A, C, D \in N, a \in T$. Assume that $\lambda \notin L(G)$. If $\lambda \in L(G)$, we directly include λ to the axiom of the contextual grammar. Now, construct a maximal semi-bracketed contextual grammar $G' = (V, \{[S]\}, P')$ where $V = \{N \cup T \cup \{\$\}\}$ and P' consists of the following productions.

1. $(\{[S]\}, \{(\$, \$AC)\})$ for each $S \rightarrow AC \in P$
2. $(\{[S]\}, \{(\$, \$a)\})$ for each $S \rightarrow a \in P$
3. $(\{[S](\$A)^*\$C\}, \{(\$, \$DE)\})$ for each $C \rightarrow DE \in P$
4. $(\{[S](\$A)^*\$C\}, \{(\$, \$a)\})$ for each $C \rightarrow a \in P$
5. $(\{[S](\$A)^*\$a[C]^+\}, \{(\$, \$DE)\})$ for each $C \rightarrow DE \in P$
6. $(\{[S](\$A)^*\$a[C]^+\}, \{(\$, \$b)\})$ for each $C \rightarrow b \in P$
7. $(\{[S]((\$A)^*\$a[C]^+\$A)\}, \{(\$, \$CD)\})$ for each $A \rightarrow CD \in P$
8. $(\{[S]((\$A)^*\$a[C]^+\$A)\}, \{(\$, \$b)\})$ for each $A \rightarrow b \in P$

The intuition behind the construction of G' is as follows. Once a variable is used (i.e., the production rule for the corresponding variable is applied), the symbol $\$$ is introduced to the

right of the used variable to identify that the variable is dead. Therefore, all the variables placed left of \$ are already used and the variables placed after the rightmost \$ indicates that the variables are alive and are not used yet. The maximality condition helps to avoid choosing the used variable again. Since the variables are used based on the leftmost derivation, the first variable after the rightmost \$ is applied in each derivation of G' . As the given grammar is in Chomsky normal form, all variables must have a production rule (no useless variables) and thus at each derivation in G' , the leftmost variable (after \$) has a rule and is processed.

We define the language for G' as $L(G') = \{pr_V(w) \mid z \xRightarrow{*}_G w, w \in (V \cup B)^*tB^+, t \in T\}$ and the projection is defined as follows:

$$pr_V(a) = \begin{cases} a, & \text{for } a \in T \\ \lambda, & \text{for } a \notin T \end{cases}$$

Since at each derivation, the leftmost non-terminal symbol is processed, a string $w \in (V \cup B)^*tB^+$, $t \in T$ is possible only when the rightmost variable symbol is processed. If w is not a string of this form, then it means that w contains some non-terminals and thus we cannot apply the projection to the string, and therefore, no string is collected in the language $L(G')$. Therefore, the strings in the language $L(G')$ is non-empty only when all the variables are replaced by a terminal symbol in the string w . \square

Now, let us verify the result with an example from a context-free grammar. Assume that a context-free grammar has the following rules in Chomsky normal form. $P = \{S \rightarrow AC, A \rightarrow DE, D \rightarrow a, E \rightarrow FF, F \rightarrow d, C \rightarrow b\}$. Consider a string $addb$ which can be obtained by the following left-most derivation $S \xRightarrow{} AC \xRightarrow{} DEC \xRightarrow{} aEC \xRightarrow{} aFFC \xRightarrow{} adFC \xRightarrow{} addC \xRightarrow{} addb$. Let us see how we can achieve this word by using the rules in P' . The number at the suffix in each derivation symbol ' $\xRightarrow{}_i$ ' indicates the rule which is applied from P' .

$$\begin{aligned} \downarrow[S]\downarrow &\xRightarrow{1} [\$^\downarrow[S]\$A^\downarrow C] \xRightarrow{3} [\$[\$^\downarrow[S]\$A\$D^\downarrow E]C] \xRightarrow{4} ([\$]^2[\$^\downarrow[S]\$A\$D\$a]E)^\downarrow C] \xRightarrow{5} \\ &([\$]^3[\$^\downarrow[S]\$A\$D\$a]E)\$F^\downarrow F]C] \xRightarrow{8} ([\$]^4[\$^\downarrow[S]\$A\$D\$a]E)\$F\$d]F]^\downarrow C] \xRightarrow{5} \\ ([\$]^4[\$^\downarrow[S]\$A\$D\$a]E)\$F\$d]F]\$d]C]^\downarrow &\xRightarrow{8} ([\$]^5[\$[S]\$A\$D\$a]E)\$F\$d]F]\$d]C]\$b] \xRightarrow{pr(V)} addb. \end{aligned}$$

Note that in the above proof, the maximality condition is important. This helps to choose the rightmost \$ everytime and avoids to choose the used variable again. Also, the condition of leftmost derivation is important here which avoids any variable before the rightmost \$ to be alive. From these details, it is easy to see that $L(G) = L(G')$, thus the family of context-free languages is subset of semi-bracketed contextual grammars with regular choice.

Corollary 5.6 $CF \subset MSBIC(REG)$

Proof. Follows from above theorem and previous lemmas. \square

6 Generative Power

As in this paper, our aim is to find the relevance of bracketed contextual grammars for natural language constraints, we do not analyze the generative power and hierarchical relations of the bracketed contextual grammars in depth. We present only a few results in this aspect and a detailed analysis is left as a future work.

Lemma 6.1 $FBIC(REG) \subset SBIC(REG)$.

Proof. The relation $FBIC(REG) \subseteq SBIC(REG)$ is obvious, since every selector in FBIC grammar is in $MDC(V)$ which can also be the case for SBIC grammar. To prove the strictness,

consider the language $L_1 = \{a^n b^m c^n d^m \mid n, m \geq 1\}$. It was shown earlier that L_1 can be generated by a MSBIC grammar with regular selector. The same grammar G_1 is applicable for SBIC grammar. However, it was proved in [16] that $L_1 \notin FBIC(REG)$. \square

The result for finite selectors is not trivial here and we give a separate proof for the case of finite selectors.

Lemma 6.2 $FBIC(FIN) \subset SBIC(FIN)$.

Proof. The part $FBIC(FIN) \subseteq SBIC(FIN)$ is obvious. To prove the strictness, let us consider a language $L_5 = \{(a^* b b a^*)^+\}$. That is, L_5 contains the strings with an even number of b 's, where any number of a can be appeared before or after two consecutive b . This language can be generated by the following semi-bracketed contextual grammar with finite choice

$$G_5 = (\{a, b\}, \{[b][b]\}, \{([b][b], \{(bb, \lambda), (\lambda, bb), (a, \lambda), (\lambda, a)\}), ((bb), (\lambda, a)), ([bb], (a, \lambda))\}).$$

A sample derivation in this mode is given by

$$\begin{aligned} \downarrow [b][b] \downarrow &\Longrightarrow [bb \downarrow [b][b] \downarrow] \Longrightarrow [bb \downarrow [b][b] \downarrow bb] \Longrightarrow [bb \downarrow [b][b] \downarrow bb] \downarrow bb] \Longrightarrow [bb \downarrow [b][b] \downarrow a] \downarrow bb] \downarrow bb] \Longrightarrow \\ &\downarrow [bb \downarrow [a[b][b] \downarrow a] \downarrow bb] \downarrow bb] \Longrightarrow [a \downarrow [bb] \downarrow [a[b][b] \downarrow a] \downarrow bb] \downarrow bb] \downarrow \Longrightarrow_{pr(V)}^* abbabbabbba. \end{aligned}$$

First, the necessary even number of b can be generated by the rule $([b][b], \{(bb, \lambda), (\lambda, bb)\})$ and then a can be inserted any number of times before or after two bb by the remaining rules. Note that the rule $([b][b], \{(a, \lambda), (\lambda, a)\})$ is necessary, since without this rule, a cannot be introduced adjacent to $[b][b]$. It is easy to see that $L(G_5) = L_5$. However, this language L_5 cannot be generated by a fully bracketed grammar with finite choice. Conversely, let us assume that G'_5 is a fully bracketed contextual grammar with finite choice that generates L_5 . Consider the strings over bb only. Since the string $(bb)^i$ is in the language for a large i , there must be a derivation such that $x_1 x_2 x_3 \Longrightarrow_{G'_5} x_1 [u x_2 v] x_3 \Longrightarrow_{G'_5}^* x_1 ([u]^k x_2 (v))^k x_3$ (by repeatedly choosing x_2 as the selector) such that $x_1 x_2 x_3$ and $x_2 \in MDC(V)$ with $x_1, x_3 \in (b \cup B)^*$, $x_2 \in (b \cup B)^+$, $uv \in b^{2j}$ for $j \geq 1$ and at least one of them is non-empty. Assume that u is a non-empty string. Since, $x_1 u^2 a^r u^{k-2} x_2 v^k x_3 \in L_5$ for a large r (as any number of a can be inserted before or after bb), there must be a selector $u^{k-2} x_2 v^{k-2} \in MDC(V)$ in G'_5 . But such a selector is not of finite length for a large k . On the other hand, if a 's are adjoined first, we get $x_1 ([a]^r x_2 ())^r x_3$. But then, $x_1 ([a]^r ([u]^{k-2} x_2 (v))^{k-2} ())^r x_3$ can only be generated, but u cannot be adjoined in between x_1 and a^r (otherwise, the selector cannot be in $MDC(V)$) and therefore $x_1 u^2 a^r u^{k-2} x_2 v^k x_3$ cannot be generated. Hence $L_5 \notin FBIC(FIN)$. Note that, since $\lambda \in MDC(V)$, a clever rule like $(\lambda, (a, \lambda))$ is also not useful for a FBIC grammar. If such a rule is there, a can be inserted in between b and b , thus a word not in the language is generated. \square

Lemma 6.3 $MSBIC(FIN) - (ICC(REG) \cup MIC(REG)) \neq \emptyset$.

Proof. Consider a language $L_6 = \{a^m b \mid m \geq 1\} \cup \{a^n b^n \mid n \geq 1\}$. This can be generated by the following semi-bracketed contextual grammar with finite choice.

$$G_6 = (\{a, b\}, \{[a][b], [ab]\}, \{([a], (a, \lambda)), ([ab], (a, b))\}).$$

A sample derivation of the grammar for L_6 is given as below.

$$\begin{aligned} \downarrow [a] \downarrow [b] &\Longrightarrow [a \downarrow [a] \downarrow \lambda] [b] \Longrightarrow [a \downarrow [a] \downarrow \lambda] [b] \Longrightarrow [a \downarrow [a] \downarrow \lambda] [b] \downarrow [b] \Longrightarrow_{pr(V)}^* a^m b. \\ \downarrow [ab] \downarrow &\Longrightarrow [a \downarrow [ab] \downarrow] \Longrightarrow [a \downarrow [ab] \downarrow] [b] \Longrightarrow [a \downarrow [a[ab] \downarrow] \downarrow] [b] \downarrow [b] \Longrightarrow_{pr(V)}^* a^n b^n. \end{aligned}$$

However, it was proved in [18] that the language L_6 can neither be generated by an internal contextual grammar nor by a maximal contextual grammar. \square

7 Conclusion

In this paper we have introduced a new class of contextual grammars namely semi-bracketed contextual grammars. Though not in detail, their relevance and suitability with natural language construction has been analyzed by incorporating the maximality feature. We have shown that context-free languages are strict subset of maximal bracketed contextual grammars with regular choice. This is an interesting result in the field of contextual grammars since no family of contextual languages, especially the class of contextual grammars which generate the non-context-free languages was shown to be a superset of context-free languages. Note that there are already some classes of contextual grammars are shown to possess the MCS formalisms without this result; for details, we refer to [10],[11]. However, the parsing issue for maximal semi-bracketed contextual grammars has not been discussed here but they are yet to be explored. We conjecture that $MSBIC(REG) \subset CS$, since every MSBIC grammar with regular choice can be transformed to an equivalent length increasing grammar. Note that we do not use (λ, λ) context in a SBIC grammar, so at each derivation, the length is increasing. To prove the strictness, we can show that there is a context-sensitive language (but which is yet to be identified) that cannot be generated by a MSBIC grammar with regular choice. We do not bother about the last condition mentioned for MCS formalism since the condition 4 is satisfied for any class of contextual grammars (the length difference between any two consecutive words in a contextual language is less than or equal to the total length of the maximal context). One natural question comes at this stage is when already several MCS formalisms exist in literature (including in the domain of contextual grammars), why we are interested in finding more formalisms? The answer to this is all the existing MCS formalisms are using variables or rewriting (in terms of symbols or trees) or not all derivation steps are accounted in the generated language. But, the MCS formalisms based on contextual grammars are *pure* grammars (no non-terminals) and all derivation steps are accounted in the language. But the previous MCS formalisms in the domain of contextual grammars do not have a structure to the strings and in this paper, we made an attempt to identify the MCS formalisms with the structure to the strings is maintained.

A small work on generative power of SBIC grammars has also been done, but a detailed study on the generative power and hierarchical relations of these grammars with the other bracketed contextual grammars and Chomsky grammars need further study. We also conjecture that $MSBIC(FIN) \subset CF$. Though the derived strings of SBIC or MSBIC grammars preserve the tree structure of $MDC(V)$ languages, the properties mentioned in [16] do not hold here. For instance, given a derivation tree for a word generated by a bracketed contextual grammar, the leaf edges identify the axiom used [16]. But that is not true for a derivation tree of a string generated by semi-bracketed contextual grammars (easy to see the difference when the selectors are not of MDC strings), thus analyzing the properties of the tree structure for semi-bracketed contextual grammars needs further study.

References

- [1] Boullier, P. Range concatenation grammars. Proceedings of *Sixth International Workshop on Parsing Technologies (IWPT '00)*. 2000, 53–64.
- [2] Boullier, P. (2001). From contextual grammars to range concatenation grammars. *Electronic Notes in Theoretical Computer Science*, **53**.
- [3] Ehrenfeucht, A., Ilie, L.; Păun, Gh.; Rozenberg, G.; Salomaa, A. On the generative capacity of certain classes of contextual grammars. In *Mathematical Linguistics and Related Topics*; Păun, Gh.; Ed.; The Publ. House of the Romanian Academy: Bucharest, 1995; pp 105–118.

- [4] Ilie, L. A non-semilinear language generated by an internal contextual grammar with finite selection. *Ann. Univ. Bucharest Math. Inform. Series.* 1996, 45/1, 63–70.
- [5] Ilie, L. On computational complexity of contextual languages. *Theo. Comp. Science.* 1997, 183/1, 33–44.
- [6] Ilie, L. Some recent results in contextual grammars. *Bull. EATCS.* 1997, 62, 172–194.
- [7] Joshi, A.K. How much context-sensitivity is required to provide structural descriptions: Tree adjoining grammars. In *Natural Language Processing: Psycholinguistic, Computational, and Theoretical Perspectives*; David, D.; Lauri, K.; Arnold, Z.; Eds.; Cambridge University Press: New York, 1985; pp 206–250.
- [8] Joshi, A.K. An introduction to tree adjoining grammars. In *Mathematics of Language*; Manaster, R.A.; Ed.; John Benjamins: Amsterdam, PH, 1987; pp 87–114.
- [9] Krishna, S.N.; Lakshmanan, K.; Rama, R. On some classes of contextual grammars. *Intern. J. of Comp. Math.* 2003, 80/2, 151–164.
- [10] Lakshmanan, K.; Krishna, S.N.; Rama, R.; Martin-Vide, C. Internal contextual grammars for mildly context sensitive languages, *Research on Language and Computation*, 2007, 5, 181–197.
- [11] Lakshmanan, K, End-marked Maximal Depth-first Contextual Grammars, Proceedings of *Developments in Lang. Theory'08*, LNCS 4036, 2008, 339-350.
- [12] Marcus, S. *Algebraic Linguistics, Analytical Models*; Academic Press: New York, 1967.
- [13] Marcus, S. Contextual grammars. *Rev. Roum. Pures. Appl.* 1969, 14, 1525–1534.
- [14] Marcus, S.; Martin-Vide, C.; Păun, Gh. On internal contextual grammars with maximal use of selectors. Proc. of *8th Conf. Automata and Formal Languages*. Salgotarjan, 1996. Also in *Publ. Math. Debrecen.* 1999, 54, 933–947.
- [15] Martin Kappes, On the Generative Capacity of Bracketed Contextual Grammars, *Grammars*, **1**, 1998, 91–101.
- [16] C. Martin-Vide, G. Păun, Structured Contextual Grammars, *Grammars*, **1**, 1998, 33–55.
- [17] Marcus, S.; Martin-vide, C.; Păun, Gh. Contextual grammars as generative models of natural languages. *Computational Linguistics.* 1998, 24(2), 245–274.
- [18] Păun, Gh. *Marcus Contextual Grammars*; Kluwer Academic Publishers: Dordrecht, The Netherland, 1997.
- [19] Păun, Gh.; Nguyen, X.M. On the inner contextual grammars. *Rev. Roum. Pures. Appl.* 1980, 25, 641–651.
- [20] Păun, Gh., Rozenberg, G., Salomaa, A.: Contextual grammars: erasing, determinism, one-side contexts, *Proc. of DLT'93*, 1993, 370–388.
- [21] Salomaa, A. *Formal Languages*. Academic Press: New York, 1973.