p ( )
URL: http://www.elsevier.nl/locate/entcs/volume53.html 12 pages

From Contextual Grammars to Range Concatenation Grammars

Pierre Boullier [1]

INRIA-Rocquencourt
Domaine de Voluceau
B.P. 105
78153 Le Chesnay Cedex, France

Abstract

Though the field of natural language processing is one of the major aims that has led to the definition of contextual grammars, very little was made on that subject. One reason is certainly the lack of efficient parsers for contextual languages. In this paper we show how some subclasses of contextual grammars can be translated into equivalent range concatenation grammars and can thus be parsed in polynomial time. However, on some other subclasses, this translation schema only succeeds if the range concatenation grammar formalism is extended. We show that the languages defined by such an extension may need an exponential parse time.

1 Introduction

Contextual Grammars (CGs) were introduced in [9] and many variants have already been investigated (see [5] for a recent survey). CGs are *pure* grammars since they do not use any auxiliary symbol: starting from some basic sentences called *axioms*, this formalism defines new sentences by inserting pair of words called *contexts* around substrings belonging to some languages called *selectors*. In this derivation process, each sentential form is in fact a sentence which contains all previously introduced terminal symbols and some new ones: the symbols of a context. CGs were mainly studied from a mathematical point of view. However, in [8], it is shown that some variants could be used in natural language processing (NLP) but the authors also indicate that one of the main

[1] Email: Pierre.Boullier@inria.fr

topics that are still poorly investigated is the parsing of contextual languages (CLs), as well as the study of their complexity.

On the other hand, range concatenation grammars (RCGs) is a syntactic formalism (see for example [4]) which defines a class of languages called *range concatenation languages* (RCLs) that exactly covers the class *PTIME* of languages recognizable in deterministic polynomial time. An interesting property of this formalism is that many grammatical formalisms used in NLP can be translated into equivalent RCGs that can be parsed very efficiently (see for example [1]). The rewriting rules of RCGs are called *clauses* and apply to composite objects named *predicates* which are nonterminal symbols with arguments. These arguments are bound to *ranges* (ordered pairs of integers which denote occurrences of substrings in a word).

The purpose of this paper is to show how some subclasses of CGs can be translated into equivalent RCGs, and can thus be parsed in polynomial time. However, for some other subclasses, this translation failed. To cope with this situation, we extend the RCG class, and we show how to translate CGs into this extended form. Unfortunately, its parsing may take an exponential time.

2 Range Concatenation Grammars

This section introduces the notion of RCG and presents some of its properties; more details appear in [4].

2.1 Positive Range Concatenation Grammars

A *positive RCG* (PRCG) $G = (N, T, V, P, S)$ is a 5-tuple where N is a finite non-empty set of *nonterminal symbols* (also called *predicate names*), T and V are finite, disjoint sets of *terminal symbols* and *variable symbols* respectively, $S \in N$ is the *axiom*, and P is a finite set of *clauses*

$$\psi_0 \rightarrow \psi_1 \dots \psi_m$$

where $m \geq 0$ and each of $\psi_0, \psi_1, \dots, \psi_m$ is a *predicate* of the form $A(\vec{\alpha})$ where $A \in N$, $\vec{\alpha}$ is a sequence $\alpha_1, \dots, \alpha_p$, of $p \geq 1$ *arguments*, p is its *arity* and each argument α_i, $1 \leq i \leq p$, is a string over $T \cup V$. In a clause, its left-hand side ψ_0 is a predicate *definition* while in its right-hand side the ψ_j's, $1 \leq j \leq m$ are predicate *calls*. If the left-hand side of a clause has the form $A(\vec{\alpha})$, we have an *A-clause*.

Each nonterminal $A \in N$ has a fixed arity whose value is *arity*(A). By definition *arity*$(S) = 1$. The *arity* k of a grammar (resulting in a k-PRCG), is the maximum arity of its nonterminals. The *size* of a clause $c = A_0(\vec{\alpha_0}) \rightarrow A_1(\vec{\alpha_1}) \dots A_m(\vec{\alpha_m})$ is the integer $|c| = \sum_{i=0}^{m} \mathit{arity}(A_i)$ and the *size* of G is $|G| = \sum_{c \in P} |c|$.

The language defined by a PRCG is based on the notion of *range*. For a given string $w = a_1 \dots a_n \in T^*$, a pair of integers (i, j) s.t. $0 \leq i \leq j \leq n$ is called a *range*, and is denoted $\langle i..j \rangle_w$. In the range $\langle i..j \rangle_w$, i is its *lower bound*,

j is its *upper bound* and $j - i$ is its *size*. If $i = j$, we have an *empty* range. For a given w, the set of all ranges is noted $\mathcal{R}_w$. In fact, $\langle i..j\rangle_w$ denotes the occurrence of the string $a_{i+1} \ldots a_j$ in w. Let $\rho = \langle i..j\rangle_w$ be a range in $\mathcal{R}_w$ for some $w = a_1 \ldots a_n \in T^*$, the substring (not the occurrence!) $a_{i+1} \ldots a_j$ of w is denoted w^ρ. More generally, if $\vec{\rho}$ denotes the sequence of p ranges $\rho_1, \ldots, \rho_p$, $\rho_k = \langle i_k..j_k\rangle_w$, $1 \leq k \leq p$, the string $w^{\rho_1} \ldots w^{\rho_p}$ is denoted $w^{\vec{\rho}}$. Two ranges $\langle i..j\rangle_w$ and $\langle k..l\rangle_w$ can be *concatenated* iff the upper bound j and the lower bound k are equal, the result is the range $\langle i..l\rangle_w \in \mathcal{R}_w$.

In any PRCG, terminals, variables and arguments of a clause denote ranges. The empty argument denotes an empty range. A terminal t denotes the range $\langle j-1..j\rangle_w$ iff $w = a_1 \ldots a_n$ and $t = a_j$. More generally, a string of the form XY denotes a range iff both X and Y denote ranges that can be concatenated: the concatenation on strings matches the concatenation on ranges.

For some $w \in T^*$, we say that $A(\rho_1, \ldots, \rho_p)$ is an *instantiation* of the predicate $A(\alpha_1, \ldots, \alpha_p)$ iff $\rho_i \in \mathcal{R}_w, 1 \leq i \leq p$ and each symbol (terminal or variable) of $\alpha_1, \ldots, \alpha_p$ denotes a range in $\mathcal{R}_w$ s.t. α_i denotes ρ_i, $1 \leq i \leq p$.

Note that, in a clause, several occurrences of the same variable always denote the same range, while several occurrences of the same terminal symbol may denote different ranges. If, in a clause, its predicates are instantiated, we have an *instantiated clause*.

For a PRCG $G = (N, T, V, P, S)$ and a string $w \in T^*$, a binary *derive* relation, denoted $\underset{G}{\Rightarrow}$, is defined on strings of instantiated predicates. If $\Gamma_1\ \gamma\ \Gamma_2$ is a string of instantiated predicates and if γ is the left-hand side of some instantiated clause $\gamma \rightarrow \Gamma$, then we write $\Gamma_1\ \gamma\ \Gamma_2 \underset{G}{\Rightarrow} \Gamma_1\ \Gamma\ \Gamma_2$.

A string w is a sentence iff we have a *complete derivation* $S(\langle 0..|w|\rangle_w) \overset{+}{\underset{G}{\Rightarrow}} \varepsilon$. The language $\mathcal{L}(G)$ defined by a PRCG G is the set of all its sentences.

2.2 Negative Range Concatenation Grammars

A *negative RCG* (NRCG) $G = (N, T, V, P, S)$ is like a PRCG, except that some predicate calls have the form $\overline{A(\alpha_1, \ldots, \alpha_p)}$.

A predicate call of the form $\overline{A(\alpha_1, \ldots, \alpha_p)}$ is said to be a *negative predicate call*. A *range concatenation grammar* (RCG) is either a PRCG or a NRCG.

In a NRCG, a negative predicate call defines the complement language (w.r.t. T^*) of its positive counterpart: an instantiated negative predicate succeeds (i.e., the derive relation $\underset{G}{\Rightarrow}$ is extended to allow $\overline{A(\vec{\rho})} \underset{G}{\Rightarrow} \varepsilon$) iff its positive counterpart (always) fails (i.e., $(A(\vec{\rho}), \varepsilon) \notin \overset{+}{\underset{G}{\Rightarrow}}$). This definition is based on a "negation by failure" rule (see [4] for a more precise discussion). However, in order to avoid inconsistencies occurring when an instantiated predicate can derive its own negative counterpart (e.g., with a clause of the form $A(X) \rightarrow \overline{A(X)}$), we prohibit *inconsistent* derivations exhibiting this possibil-

ity.

2.3 Parse Time Complexity

In [4], we presented a parsing algorithm which, for any RCG G and any input string of length l, produces its parse forest in $\mathcal{O}(|G|l^d)$ time. The *degree* d_c of a clause c is its number of free (independent) bounds, and the *degree* d of a grammar G is the maximum value of all d_c's.

3 Contextual Grammars

A *contextual grammar* with choice (CG), is a tuple $K = (T, A, (S_1, C_1), \ldots, (S_n, C_n))$, $n \geq 1$, where T is a finite set of terminal symbols, A is a finite language over T whose elements are called *axioms*, $S_1, \ldots, S_n$ are languages over T called *selectors*,[2] and $C_1, \ldots, C_n$ are finite subsets of $T^* \times T^*$, the elements of which, written in the form (u, v), are called *contexts*. If the selectors $S_1, \ldots, S_n$ are all languages in a given family $\mathcal{F}$, we say that K is a CG with $\mathcal{F}$ choice.

For a given CG K, we can define either an *external derive* relation denoted $\underset{K,ex}{\Rightarrow}$ or an *internal derive* relation denoted $\underset{K,in}{\Rightarrow}$. We write $x \underset{K,ex}{\Rightarrow} y$ iff $y = uxv$ for $(u, v) \in C_i$, $x \in S_i$, for some i, $1 \leq i \leq n$ and we write $x \underset{K,in}{\Rightarrow} y$ iff $x = x_1x_2x_3$, $y = x_1ux_2vx_3$ for $(u, v) \in C_i$, $x_2 \in S_i$, for some i, $1 \leq i \leq n$. If α is a derive relation mode (i.e., $\alpha \in \{ex, in\}$), the language defined by K w.r.t. α is the *contextual language* (CL) $\mathcal{L}_\alpha(K)$ defined by $\{w \in T^* \mid z \overset{*}{\underset{K,\alpha}{\Rightarrow}} w, \text{ for some } z \in A\}$.

Let $K = (T, A, (S_1, C_1), \ldots, (S_n, C_n))$ be a CG with $\mathcal{F}$ choice used either in external or internal mode. If, from K, we want to build an equivalent (self-contained) RCG, the first criterion we have to meet is that each member of the family of languages $\mathcal{F}$ must be an RCL. Thus in the sequel we will assume that each selector $S_i \in \{S_1, \ldots, S_n\}$ is an RCL, defined by some (usually unspecified) RCG, the axiom of which is, by definition, the nonterminal written $[S_i]$. In other words, each occurrence of a predicate call of the form $[S_i](\alpha)$, for some string w, is true iff α denotes some range $\rho \in \mathcal{R}_w$ s.t. the string w^ρ is a sentence of the selector language S_i.

3.1 From External CGs to RCGs

For each CG $K = (T, A, (S_1, C_1), \ldots, (S_n, C_n))$ with external derivation mode which defines the language $\mathcal{L}_{ex}(K)$, we build an RCG $G = (N, T, V, P, S)$ s.t. $N = \{S, [S_1], \ldots, [S_n]\} \cup \{[S_1, C_1], \ldots, [S_n, C_n]\}$, $V = \{X\}$,[3] and the set

[2] Note that the formalism in which the selectors are defined is not specified.

[3] In the sequel, without any further explanations, variables will be denoted by (subscripted) late occurring upper-case letters such as X, Y, Z.

of clauses P is defined by the three clause schemata

$$\begin{aligned} 1 &: S(z) && \rightarrow \varepsilon \\ 2 &: S(X) && \rightarrow [S_i, C_i](X) \\ 3 &: [S_i, C_i](uXv) && \rightarrow [S_i](X)\ S(X) \end{aligned}$$

The first clause schema applies $\forall z \in A$, the second $\forall i \in \{1, \ldots, n\}$ and the third $\forall i \in \{1, \ldots, n\}, \forall (u, v) \in C_i$.

Note that the size of G is linear in the size of K.

By induction on the length of derivations, we can show that $\mathcal{L}_{ex}(K) = \mathcal{L}(G)$.

Now, if we look at the parse time complexity of G, we see that the clauses generated by the three clause schemata can be parsed in time quadratic in the length l of the input string since their degree is two. Thus the total parse time of G is $\max(\mathcal{O}(l^2), \mathcal{O}(l^{d_i}))$, if we assume that, for any selector language S_i, the membership problem can be solved in $\mathcal{O}(l^{d_i})$ time. If $d = \max_{i=1}^{n} d_i$ is the maximum degree of the selector languages, this shows that any external CL can be parsed in $\mathcal{O}(l^{max(2,d)})$ time.

3.2 From Internal CGs to RCGs

This type of CGs is interesting because it has been shown (see for example [8]) that the three basic non context-free constructions of NLs,[4] upon which the notion of mild context sensitivity (MCS) (see [7]) is built, can be defined with internal CGs with regular choice.

In order to build an equivalent RCG from an internal CG, it is tempting to mimic the case of external CGs and to provide the same kind of clause schemata as the ones used in Section 3.1. Doing that, the first two clause schemata stay unchanged while the third one would be changed into

$$3' : [S_i, C_i](X_1 u X_2 v X_3) \rightarrow [S_i](X_2)\ \ S(X_1 X_2 X_3)$$

$\forall i \in \{1, \ldots, n\}, \forall (u, v) \in C_i$.

However, this translation schema is erroneous since the clauses of the form $3'$ can only be instantiated iff $X_1 X_2 X_3$, the argument of the rightmost predicate call, denotes a range, that is iff the three ranges bounded respectively to X_1, X_2 and X_3 can be concatenated. This is possible iff $X_1 u X_2 v X_3 = X_1 X_2 X_3$ (i.e., $uv = \varepsilon$).

Since, within the (standard) RCG formalism, it is not possible to express the fact that the input string is changed during a derivation, we propose an extension of RCGs called *dynamic* RCGs (DRCGs) in which this operation is allowed. A DRCG is an RCG in which a specific nonterminal named *catenate* is predefined.

[4] That is, multiple agreement, cross agreement and duplication, respectively abstracted by the languages $\{a^n b^n c^n \mid n \geq 1\}$, $\{a^n b^m c^n d^m \mid m, n \geq 1\}$ and $\{wcw \mid w \in \{a, b\}^*\}$.

All *catenate* predicates have the form $\psi = catenate(A, \vec{\alpha})$ for positive predicate calls or the form $\psi = \overline{catenate(A, \vec{\alpha})}$ for negative predicate calls, where $A \in N$, $arity(A) = 1$ and $\vec{\alpha} \in ((V \cup T)^*)^+$.[5] Let $\gamma = catenate(A, \vec{\rho})$, $\vec{\rho} \in \mathcal{R}_w^p$ be an instantiation of ψ for some $w \in T^*$, let w' be the string $w^{\vec{\rho}}$, and let ρ' be the range $\langle 0..|w'| \rangle_{w'}$. If $\Gamma_1\Gamma_2$ is a string of instantiated predicates, we extend the derive relation $\underset{G}{\Rightarrow}$: we write $\Gamma_1\gamma\Gamma_2 \underset{G}{\Rightarrow} \Gamma_1\ A(\rho')\ \Gamma_2$ when ψ is a positive *catenate* call or $\Gamma_1\gamma\Gamma_2 \underset{G}{\Rightarrow} \Gamma_1\Gamma_2$ when ψ is a negative *catenate* call s.t. $(A(\rho'), \varepsilon) \notin \overset{+}{\underset{G}{\Rightarrow}}$.

Thus, a *catenate* call allows to dynamically change the "input" string during a derivation, its size can even increase (e.g., consider the effect of the clause $S(X) \rightarrow catenate(S, X, X)$).

Of course, this extension of the RCG formalism is not harmless on parse time complexities, even if we restrict ourselves to bounded DRCGs.[6] If we assume that the length of any intermediate input string is bounded by l, which is the case of CGs, the number of these strings is $\mathcal{O}((|T| + 1)^l)$. Thus, even in that case, the parse time of 1-bounded DRCLs increases from polynomial to exponential.[7]

Now, we are able to transform any internal CG into a DRCG by changing the third clause schema of Section 3.1 into

$$3'' : [S_i, C_i](X_1 u X_2 v X_3) \rightarrow [S_i](X_2)\ \ catenate(S, X_1, X_2, X_3)$$

The corresponding DRCG is 1-bounded and its language can be parsed at worst in exponential time, for any RCL choice. We can show that $\mathcal{L}_{in}(K) = \mathcal{L}(G'')$, if G'' is the DRCG derived by the rule schemata 1, 2 and $3''$.

4 Maximal Use of Selectors

The (potential) adequacy of internal CGs to NLP has been studied in [8], where two variants of the relation $\underset{K,in}{\Rightarrow}$ are defined. These variants use selectors in a maximal sense: a context is adjoined to a word-selector if this word is the largest on that place (no other word containing it as a proper sub-word can be a word-selector). The purpose of this section is to show how these two variants can also be translated into equivalent DRCGs.

[5] In fact, it is possible to define a generalization of *catenate* which also works for non-unary nonterminals.

[6] A DRCG is said to be *c-bounded* if their exists a constant c s.t. for any initial input string w of length l, the size of any range (i.e., the length of any intermediate input string) is less than or equal to $c * l$.

[7] The parse time of DRCLs stays polynomial for the subclasses in which it can be shown that the number of dynamic intermediate strings is itself polynomial.

4.1 Maximal Local Mode

In this first variant a derivation is in the *maximal local mode*, and we write $\underset{K,Ml}{\Rightarrow}$ for the corresponding derive relation, iff $x = x_1x_2x_3$, $y = x_1ux_2vx_3$, for $x_2 \in S_i$, $(u, v) \in C_i$, for some $1 \leq i \leq n$, and there are no x'_1, x'_2, $x'_3 \in T^*$ s.t. $x = x'_1x'_2x'_3$, $x'_2 \in S_i$, and $|x'_1| \leq |x_1|$, $|x'_3| \leq |x_3|$, $|x'_2| > |x_2|$. In that case, the word-selector x_2 is maximal in S_i.

The translation of a CG $K = (T, A, (S_1, C_1), \ldots, (S_n, C_n))$ with internal derivations in maximal local mode into an equivalent DRCG only differs from the schema proposed in Section 3.2 in the processing, by means of RCG clauses, of the maximal local mode constraint.

The clause schema $3''$ is changed into

$$3''' : [S_i, C_i](X_1uX_2vX_3) \rightarrow \overline{[S_i]_{Ml}(X_1uX_2vX_3, uX_2v, X_2)}$$
$$[S_i](X_2)\ \mathit{catenate}(S, X_1, X_2, X_3)$$

in which the ternary nonterminal $[S_i]_{Ml}$, $1 \leq i \leq n$ checks the maximal local mode constraint. All predicates of the form $[S_i]_{Ml}(Y, X'_2, X_2)$ are s.t. X_2 is a subrange of X'_2, and X'_2 is itself a subrange of Y. Let x_2 be the string selected by X_2 and let ux_2v be the string selected by X'_2 for some $u, v \in T^*$, and let $y = x_1x'_1ux_2vx'_3x_3$ be the string selected by Y for some $x_1, x'_1, x'_3, x_3 \in T^*$. In that case, the predicate call $[S_i]_{Ml}(Y, X'_2, X_2)$ is true iff we have $x'_1x'_3 \neq \varepsilon$ and $x'_1x_2x'_3 \in S_i$, that is the word-selector x_2 is not maximal in S_i, and the negative call $\overline{[S_i]_{Ml}(Y, X'_2, X_2)}$ succeeds iff the word-selector x_2 is maximal in S_i. Thus, $\forall i \in \{1, \ldots, n\}$, each $[S_i]_{Ml}$ can be defined by the clause schema

$$4 : [S_i]_{Ml}(X_1X'_1X'_2X'_3X_3, X'_2, X_2) \rightarrow \overline{\mathit{null}_2(X'_1, X'_3)}$$
$$\mathit{catenate}([S_i], X'_1, X_2, X'_3)$$

The binary nonterminal $null_2$ is true iff its arguments denote empty ranges

$$\mathit{null}_2(\varepsilon, \varepsilon) \rightarrow \varepsilon$$

4.2 Maximal Global Mode

In this second variant a derivation is in the *maximal global mode*, and we write $\underset{K,Mg}{\Rightarrow}$ for the corresponding derive relation, iff $x = x_1x_2x_3$, $y = x_1ux_2vx_3$, for $x_2 \in S_i$, $(u, v) \in C_i$, for some $1 \leq i \leq n$, and there are no x'_1, x'_2, $x'_3 \in V^*$ s.t. $x = x'_1x'_2x'_3$, $x'_2 \in S_j$, for some $1 \leq j \leq n$, and $x'_1 \leq |x_1|$, $|x'_3| \leq |x_3|$, $|x'_2| > |x_2|$. In that case, the word-selector x_2 is maximal w.r.t. all selectors $S_1, \ldots, S_n$.

For the translation of an internal CG $K = (T, A, (S_1, C_1), \ldots, (S_n, C_n))$ in maximal global mode into an equivalent DRCG, the clause schema $3'''$ is

changed into

$$3'''' : [S_i, C_i](X_1 u X_2 v X_3) \rightarrow \overline{Mg(X_1 u X_2 v X_3, u X_2 v, X_2)}$$
$$[S_i](X_2)\ catenate(S, X_1, X_2, X_3)$$

where the nonterminal Mg defines the maximal global mode constraint. For each $i \in \{1, \ldots, n\}$, this constraint can be defined by the following Mg-clauses

$$Mg(X_1, X_2, X_3) \rightarrow [S_i]_{Ml}(X_1, X_2, X_3)$$

4.3 An Example

In this section, we illustrate our transformation schema on an example. In [8], it is shown that the multiple agreement property, abstracted by the language $L = \{a^n b^n c^n \mid n \geq 1\}$, is a CL that can be defined by an internal CG $K = (\{a, b, c\}, \{abc\}, (b^+, \{(a, bc)\})$, in maximal local or maximal global mode. Since in K there is only a single selector language, the local and global maximal modes are identical. When applied to K, the transformation schema of Section 4.1 gives an equivalent DRCG G whose clauses are

$$\begin{aligned}
S(abc) &\rightarrow \varepsilon \\
S(X) &\rightarrow [b^+, \{(a, bc)\}](X) \\
[b^+, \{(a, bc)\}](X_1 a X_2 bc X_3) &\rightarrow \overline{[b^+]_{Ml}(X_1 a X_2 bc X_3, a X_2 bc, X_2)} \\
&\quad [b^+](X_2)\ catenate(S, X_1, X_2, X_3) \\
[b^+]_{Ml}(X_1 X_1' X_2' X_3' X_3, X_2', X_2) &\rightarrow \overline{null_2(X_1', X_3')} \\
&\quad catenate([b^+], X_1', X_2, X_3') \\
null_2(\varepsilon, \varepsilon) &\rightarrow \varepsilon \\
[b^+](b) &\rightarrow \varepsilon \\
[b^+](bX) &\rightarrow [b^+](X)
\end{aligned}$$

Note that the last two clauses define the selector language b^+.

Below, we show how the sentence $w = aaabbbccc$ can be derived by both K and G.

Using K, we have $a\underline{b}c \underset{K,Ml}{\Rightarrow} aa\underline{bb}cc \underset{K,Ml}{\Rightarrow} aaabbbccc$ in which at each step the word-selector has been underlined.

If we consider DRCG derivations, starting from the instantiation $S(\langle 0..9\rangle_w)$ of the axiom S on w, we can build the complete derivation

$$\begin{aligned}
&\overset{2}{\underset{G}{\Rightarrow}}\ [b^+, \{(a, bc)\}](\langle 0..9\rangle_w) \\
&\overset{3'''}{\underset{G}{\Rightarrow}}\ \overline{[b^+]_{Ml}(\langle 0..9\rangle_w, \langle 2..7\rangle_w, \langle 3..5\rangle_w)} \\
&\quad [b^+](\langle 3..5\rangle_w)\ catenate(S, \langle 0..2\rangle_w, \langle 3..5\rangle_w, \langle 7..9\rangle_w)
\end{aligned}$$

$$\stackrel{+}{\underset{G}{\Rightarrow}} S(\langle 0..6\rangle_{w'=aabbcc})$$
$$\stackrel{2}{\underset{G}{\Rightarrow}} [b^+, \{(a, bc)\}](\langle 0..6\rangle_{w'})$$
$$\stackrel{3'''}{\underset{G}{\Rightarrow}} \overline{[b^+]_{Ml}(\langle 0..6\rangle_{w'}, \langle 1..5\rangle_{w'}, \langle 2..3\rangle_{w'})}$$
$$[b^+](\langle 2..3\rangle_{w'})\ \mathit{catenate}(S, \langle 0..1\rangle_{w'}, \langle 2..3\rangle_{w'}, \langle 5..6\rangle_{w'})$$
$$\stackrel{+}{\underset{G}{\Rightarrow}} S(\langle 0..3\rangle_{w''=abc})$$
$$\stackrel{1}{\underset{G}{\Rightarrow}} \varepsilon$$

which shows that $w \in \mathcal{L}(G)$.[8]

5 Other Local Variants

Two local variants of internal CG, defined in [6], are worth considering. They are, together with external CGs, the only classes of internal CGs for which complexity results are known: their membership problems can both be solved in polynomial time. Furthermore, the variant with a maximal mode of derivation can define the three basic MCS constructions. In this section, we show how these grammars can be translated into equivalent RCGs. Of course these (non dynamic) RCGs can be parsed in polynomial time.

In the first local variant, for any CG $K = (T, A, (S_1, C_1), \ldots, (S_n, C_n))$, Ilie defines a *local* derive relation, denoted $\underset{K,in,loc}{\Rightarrow}$, as follows.

If $\underset{K,in}{\Rightarrow}$ is the usual internal derive relation of Section 3 and if we consider two strings $z, x \in T^*$, s.t. $z \underset{K,in}{\Rightarrow} x$, $z = z_1z_2z_3$, $z_2 \in S_i$, $(u, v) \in C_i$, for some $1 \leq i \leq n$, and $x = z_1uz_2vz_3$, we have $x \underset{K,in,loc}{\Rightarrow} y$, w.r.t. $z \underset{K,in}{\Rightarrow} x$, and we write $z \underset{K,in}{\Rightarrow} x \underset{K,in,loc}{\Rightarrow} y$, iff we have $u = u_1u_2$, $v = v_1v_2$, $x = x_1x_2x_3$ for $x_1 = z_1u_1$, $x_2 = u_2z_2v_1$, $x_3 = v_2z_3$, $x_2 \in S_j$, $(w, s) \in C_j$, for some $1 \leq j \leq n$, and $y = x_1wx_2sx_3$.

This derive relation is called *local* because the new locations where contexts are introduced are inside (or at most adjacent to) the previously introduced context.

The second local variant is a restriction of $\underset{K,in,loc}{\Rightarrow}$ called *maximal local* and denoted $\underset{K,in,Mloc}{\Rightarrow}$. We have $z \underset{K,in}{\Rightarrow} x \underset{K,in,Mloc}{\Rightarrow} y$ iff $z \underset{K,in}{\Rightarrow} x \underset{K,in,loc}{\Rightarrow} y$ and there is no other local derivation $z \underset{K,in}{\Rightarrow} x \underset{K,in,loc}{\Rightarrow} y'$ s.t. the decomposition used for x, say $x = x'_1x'_2x'_3$, $x'_2 \in S_j$, verifies $|x'_1| \leq |x_1|$, $|x'_2| > |x_2|$ and $|x'_3| \leq |x_3|$.

[8] $\Gamma \stackrel{i}{\underset{G}{\Rightarrow}} \Gamma'$ means that this derivation step uses the clause schema i. Moreover, the reader is invited to check that, for example, the third derivation step results from the fact that we have both $([b^+]_{Ml}(\langle 0..9\rangle_w, \langle 2..7\rangle_w, \langle 3..5\rangle_w), \varepsilon) \not\in \stackrel{+}{\underset{G}{\Rightarrow}}$ and $[b^+](\langle 3..5\rangle_w) \stackrel{+}{\underset{G}{\Rightarrow}} \varepsilon$.

For $\alpha \in \{loc, Mloc\}$ and $x, y \in T^*$, we write $x \overset{*}{\underset{K,in,\alpha}{\Rightarrow}} y$ iff we have a finite derivation, in which each step, excepting the first one, is performed in α mode w.r.t. the previous one, that is

$$x = x_0 \underset{K,in}{\Rightarrow} x_1 \underset{K,in,\alpha}{\Rightarrow} x_2 \underset{K,in,\alpha}{\Rightarrow} \ldots \underset{K,in,\alpha}{\Rightarrow} x_k = y$$

for some $k \geq 0$, and, in the case $\alpha = Mloc$, the first step $x_0 \underset{K,in}{\Rightarrow} x_1$ must be performed in the maximal mode.

5.1 From Local Variant Internal CG to RCG

The translation of the CG $K = (T, A, (S_1, C_1), \ldots, (S_n, C_n))$ used with the derive relation $\underset{K,in,loc}{\Rightarrow}$, into an equivalent RCG can be performed as follows.

First, $\forall u \in A$, we specify that each axiom is a sentence by

$$S(u) \rightarrow \varepsilon$$

Second, for each S_i, $1 \leq i \leq n$, and for each axiom $u = u_1 u_2 u_3 \in A$, $u_1, u_2, u_3 \in T^*$, we statically check if u_2 is an element of S_i. If it is the case, we generate a clause of the form

$$5 : S(u_1 X U_2 Y u_3) \rightarrow \mathit{fix}(u_2, U_2)\ [S_i, C_i](X U_2 Y, U_2)$$

which specifies that the initial derivation step must be performed by $\underset{K,in}{\Rightarrow}$. The predicate name *fix*, defined by

$$\mathit{fix}(X, X) \rightarrow \varepsilon$$

is used to "anchor" a terminal string (here the occurrence of u_2, between X and Y in clause 5) at a given position by means of a variable (here U_2).

Third, for each i, $1 \leq i \leq n$ and for each $(u, v) \in C_i$ s.t. $u_1 u_2 = u, v_1 v_2 = v$, we generate

$$6 : [S_i, C_i](u_1 X u_2 Z v_1 Y v_2, Z) \rightarrow [S_i](Z)\ [S, C](X u_2 Z v_1 Y, u_2 Z v_1)$$

which processes the derivation steps in *loc* mode.

Finally, the predicate name $[S, C]$ is defined by the clauses

$$[S, C](X, Z) \rightarrow [S_i, C_i](X, Z)\ \forall i \in \{1, \ldots, n\}$$
$$[S, C](Z, Z) \rightarrow \varepsilon$$

which indicate that each derivation step, excepting the first one, is performed by a pair (S_i, C_i), until completion.

Let us assume that the selectors S_i of K are RCLs, and let L be the language defined by K with $\underset{K,in,loc}{\Rightarrow}$. We can show that the previous PRCG also defines L and, moreover, that its sentences can be parsed in time $\mathcal{O}(l^6)$ (considering the clause schema 6, we see that the degree of this PRCG is six), if we exclude the recognition time of the selector strings. Thus, if the

parse time recognition of the selector strings is $\mathcal{O}(l^d)$, L can be parsed in time $\mathcal{O}(l^{\max(6,d)})$. In [2] we show that regular, linear and context-free languages can be parsed respectively in time linear, quadratic and cubic by an equivalent PRCG and in [3], we show that any tree-adjoining language can be parsed in $\mathcal{O}(l^6)$ time by an equivalent PRCG. This shows that the languages defined by this local variant of CGs can be parsed in $\mathcal{O}(l^6)$ time if the selectors are tree-adjoining languages.

5.2 From Maximal Local Variant Internal CG to RCG

The translation process of a CG $K = (T, A, (S_1, C_1), \ldots, (S_n, C_n))$ used with the derive relation $\underset{K,in,Mloc}{\Rightarrow}$, into an equivalent RCG is similar to the previous one, except on two points. First the clauses of the form 5 are generated only if they fulfill the maximal mode condition. Second, the clause schema 6 is changed into

$$[S_i, C_i](U_1 X u_2 Z v_1 Y V_2, Z) \rightarrow \mathit{fix}(u_1, U_1)\ \mathit{fix}(v_2, V_2)\ [S_i](Z)\ [S, C](X u_2 Z v_1 Y, u_2 Z v_1)$$
$$\overline{Mloc(U_1 X u_2 Z v_1 Y V_2, Z, U_1, u_2 Z v_1, V_2)}$$

The nonterminal $Mloc$ checks that the maximal mode condition is fulfilled. [9] For each i, $1 \leq i \leq n$ and for each $(u, v) \in C_i$ s.t. $u_1 u_2 = u, v_1 v_2 = v$, we generate

$$Mloc(u_1 X u_2 Z v_1 Y v_2, Z, u_1 X_1, X_2, X_3 v_2) \rightarrow [S_i, C_i](u_1 X u_2 Z v_1 Y v_2, Z)\ gtr(u_2 Z v_1, X_2)$$

The maximal parse time complexity is reached for the $Mloc$-clauses of degree eight. This shows that the languages defined by this maximal local variant of CGs can be parsed in $\mathcal{O}(l^8)$ time if the selectors are tree-adjoining languages.

6 Conclusion

In this paper we have shown how any external CG with RCL choice, after translation into an equivalent RCG, can be parsed in polynomial time. We have also shown how some local variants of internal CGs with RCL choices can be parsed in polynomial time by equivalent RCGs. However, in order to process the general case of internal CGs and internal CGs with a maximal use of selectors, both in local or global mode, we have extended the RCG formalism and defined DRCGs. With this new formalism we have shown how CLs can be defined, but the corresponding parsing time can be exponential.

[9] A predicate call of the form $gtr(X, Y)$ simply checks that the size of the range X is greater than the size of the range Y.

Furthermore, this RCG-based CL parsing allows to process in a unified way both the context insertion phase and the word selection phase. In other words, CGs are a kind of two-level grammars since some other mechanism is needed to define the selector languages. At least for selector languages which are [D]RCLs, the translation of CGs into equivalent [D]RCGs leads to a single definition formalism.

Finally, one can note that the translation of CGs into equivalent [D]RCGs naturally provides a structure (the parse forest output by the [D]RCG parsers) for any sentence. However, the relevance of this structure in NLP would have to be demonstrated. [10]

References

[1] F. Barthélemy, P. Boullier, Ph. Deschamp, and É. de la Clergerie. 2001. Guided parsing of range concatenation languages. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL'01)*, University of Toulouse, France.

[2] P. Boullier. 2000a. A cubic time extension of context-free grammars. *Grammars*, 3(2/3):111–131.

[3] P. Boullier. 2000b. On tag parsing. *Traitement Automatique des Langues (t.a.l.)*, 41(3):111–131.

[4] P. Boullier. 2000c. Range concatenation grammars. In *Proceedings of the Sixth International Workshop on Parsing Technologies (IWPT 2000)*, pages 53–64, Trento, Italy.

[5] A. Ehrenfeucht, Gh. Păun, and G. Rozenberg. 1997. Contextual grammars. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*. Springer, Berlin.

[6] L. Ilie. 1997. On computational complexity of contextual languages. *Theoretical Computer Science*, 183(1):33–44.

[7] A. K. Joshi, 1985. *How much context-sensitivity is necessary for characterizing structural descriptions — Tree Adjoining Grammars*. Cambridge University Press, New-York, NY. D. Dowty, L. Karttunen, and A. Zwicky (eds.).

[8] S. Marcus, C. Martín-Vide, and Gh. Păun. 1998. Contextual grammars as generative models of natural languages. *Computational Linguistics*, 24(2):245–274.

[9] S. Marcus. 1969. Contextual grammars. *Revue Roumaine des Mathématiques Pures et Appliquées*, 14(10):1525–1534.

[10] A discussion on the attempts to associate a pertinent structure to CLs can be found in [8].