*p ( )*

*URL:* [*http://www.elsevier.nl/locate/entcs/volume53.html*](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 eld of natural language processing is one of the major aims that has led to the de nition of contextual grammars, very little was made on that subject. One reason is certainly the lack of e cient 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 de ned 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 de nes 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*](mailto:Pierre.Boullier@inria.fr)

*c 2001 Published by Elsevier Science B. V. Open access under* [*CC BY-NC-ND license.*](http://creativecommons.org/licenses/by-nc-nd/3.0/)

*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 syntac-* tic formalism (see for example [4]) which de nes 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 e ciently (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 nite non-empty set of nonterminal symbols (also called predicate names), T and V are nite, disjoint sets of terminal symbols and variable symbols respectively, S 2 N is the axiom, and P is a nite set of clauses*

*0 ! 1 ::: m*

*where m 0 and each of 0 ; 1 ;::: ; m is a predicate of the form A( ~ ) where*

*A 2 N , ~ is a sequence 1 ;::: ; p, of p 1 arguments, p is its arity and each*

*argument i, 1 i p, is a string over T [ V . In a clause, its left-hand side*

*0 is a predicate de nition while in its right-hand side the j's, 1 j m are predicate calls. If the left-hand side of a clause has the form A( ~ ), we have an A-clause.*

*Each nonterminal A 2 N has a xed arity whose value is arity(A). By* de nition 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 = A0 ( ~0) !

*A ( ~ ) ::: A ( ~*

*) is the integer jcj = Pm arity(A ) and the size of G is*

*1 1 m m*

*i=0 i*

*jGj = Pc2P jcj.*

*The language de ned by a PRCG is based on the notion of range. For a given string w = a1 ::: an 2 T , a pair of integers (i; j) s.t. 0 i j n is called a range, and is denoted hi::jiw. In the range hi::jiw, 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 Rw. In fact, hi::jiw denotes the occurrence of the string ai+1 ::: aj in w. Let = hi::jiw be a range in Rw for some w = a ::: a 2 T , the substring (not the occurrence!) a ::: a of w is denoted w. More generally, if ~ denotes the sequence of p ranges 1;::: ; p,*

*1 n i+1 j*

*k = hik::jkiw, 1 k p, the string w1 ::: wp is denoted w ~. Two ranges*

*hi::jiw and hk::liw can be concatenated i the upper bound j and the lower bound k are equal, the result is the range hi::liw 2 Rw.*

*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 hj 1::jiw i w = a1 ::: an and t = aj. More generally, a string of the form XY denotes a range i both X and Y denote ranges that can be concatenated: the concatenation on strings matches the concatenation on ranges.

*For some w 2 T , we say that A( ;::: ; ) is an instantiation of the*

*1 p*

*predicate A( 1;::: ; p) i i 2 Rw; 1 i p and each symbol (terminal or*

*variable) of 1;::: ; p denotes a range in Rw s.t. i denotes i, 1 i 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 di erent 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 2 T , a binary derive* relation, denoted ), is de ned on strings of instantiated predicates. If 1 2

*G*

*is a string of instantiated predicates and if is the left-hand side of some*

*instantiated clause ! , then we write 1 2 ) 1 2.*

*G*

*+*

*A string w is a sentence i we have a complete derivation S(h0::jwjiw) ) ".*

*G*

*The language L(G) de ned 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 A( 1;::: ; p).*

*A predicate call of the form A( 1;::: ; 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 de nes the complement language* (w.r.t. T ) of its positive counterpart: an instantiated negative predicate suc- ceeds (i.e., the derive relation ) is extended to allow A( ~) ) ") i its pos-

*G G*

*+*

*itive counterpart (always) fails (i.e., (A( ~); ") 62)). This de nition is based*

*G*

*on a \negation by failure" rule (see [4] for a more precise discussion). How-*

*ever, in order to avoid inconsistencies occurring when an instantiated predi-* cate can derive its own negative counterpart (e.g., with a clause of the form A(X) ! 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 O(jGjld) time. The degree dc 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 dc's.

*3 Contextual Grammars*

*A contextual grammar with choice (CG), is a tuple K = (T; A; (S1; C1);::: ; (Sn; Cn)), n 1, where T is a nite set of terminal symbols, A is a nite language over T whose elements are called axioms, S1;::: ; Sn are languages*

*over T called selectors, 2 and C ;::: ;C are nite subsets of T T , the ele-*

*1 n*

*ments of which, written in the form (u; v), are called contexts. If the selectors*

*S1;::: ; Sn are all languages in a given family F, we say that K is a CG with* F choice.

*For a given CG K, we can de ne either an external derive relation denoted*

*) or an internal derive relation denoted ) . We write x ) y i y = uxv*

*K;ex*

*K;in*

*K;ex*

*for (u; v) 2 Ci, x 2 Si, for some i, 1 i n and we write x )*

*K;in*

*y i*

*x = x1 x2 x3, y = x1 ux2vx3 for (u; v) 2 Ci, x2 2 Si, for some i, 1 i n. If is a derive relation mode (i.e., 2 fex; ing), the language de ned by K*

*w.r.t. is the contextual language (CL) L*

*w; for some z 2 Ag.*

*(K) de ned by fw 2 T j z*

*)*

*K;*

*Let K = (T; A; (S1; C1);::: ; (Sn; Cn)) be a CG with F choice used either* in external or internal mode. If, from K, we want to build an equivalent (self- contained) RCG, the rst criterion we have to meet is that each member of the family of languages F must be an RCL. Thus in the sequel we will assume that each selector Si 2 fS1;::: ; Sng is an RCL, de ned by some (usually un- speci ed) RCG, the axiom of which is, by de nition, the nonterminal written [Si]. In other words, each occurrence of a predicate call of the form [Si]( ), for some string w, is true i denotes some range 2 Rw s.t. the string w is a sentence of the selector language Si.

*3.1 From External CGs to RCGs*

*For each CG K = (T; A; (S1; C1);::: ; (Sn; Cn)) with external derivation mode which de nes the language Lex(K), we build an RCG G = (N; T; V; P; S)*

*s.t. N = fS; [S ];::: ; [S ]g[ f[S ;C ];::: ; [S ;C ]g, V = fXg, 3 and the set*

*1 n 1 1 n n*

*2 Note that the formalism in which the selectors are de ned is not speci ed.*

*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 de ned by the three clause schemata*

*1 : S(z) ! "*

*2 : S(X) ! [Si; Ci](X)*

*3 : [Si; Ci](uXv) ! [Si](X) S(X)*

*The rst clause schema applies 8z 2 A, the second 8i 2 f1;::: ; ng and the third 8i 2 f1;::: ; ng; 8(u; v) 2 Ci.*

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

*By induction on the length of derivations, we can show that Lex(K) =* 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(O(l2); O(ldi )), if we assume that, for any selector language

*Si, the membership problem can be solved in O(ldi ) time. If d = maxn di is*

*i=1*

*the maximum degree of the selector languages, this shows that any external* CL can be parsed in O(lmax(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 de ned 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 rst two clause schemata stay unchanged while the third one would be changed into

*30 : [S ;C ](X uX vX ) ! [S ](X ) S(X X X )*

*i i 1 2 3 i 2 1 2 3*

*8i 2 f1;::: ; ng; 8(u; v) 2 Ci.*

*However, this translation schema is erroneous since the clauses of the form*

*30 can only be instantiated i X X X , the argument of the rightmost predi-* cate call, denotes a range, that is i the three ranges bounded respectively to X1, X2 and X3 can be concatenated. This is possible i X1uX2vX3 = X1X2X3 (i.e., uv = ").

*1 2 3*

*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 speci c nonterminal named catenate is prede ned.

*4 That is, multiple agreement, cross agreement and duplication, respectively abstracted by the languages fan bn cn j n 1g, fan bm cn dm j m; n 1g and fwcw j w 2 fa; bg g.*

*All catenate predicates have the form = catenate(A; ~ ) for positive predicate calls or the form = catenate(A; ~ ) for negative predicate calls,*

*where A 2 N , arity(A) = 1 and ~ 2 ((V [ T ) )+. 5 Let = catenate(A; ~),*

*~ 2 Rp be an instantiation of for some w 2 T , let w0 be the string w ~, and*

*w*

*let 0 be the range h0::jw0ji 0 . If is a string of instantiated predicates,*

*w 1 2*

*2*

*we extend the derive relation ): we write 1 2*

*G*

*) 1*

*G*

*A( 0 )*

*when is*

*a positive catenate call or 1 2 ) 1 2 when is a negative catenate call*

*G*

*+*

*s.t. (A( 0 );"*

*) 62).*

*G*

*Thus, a catenate call allows to dynamically change the \input" string*

*during a derivation, its size can even increase (e.g., consider the e ect of* the clause S(X) ! 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 O((jT j + 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

*300 : [S ;C ](X uX vX ) ! [S ](X ) catenate(S; X ;X ;X )*

*i i 1 2 3 i 2*

*1 2 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 Lin(K) = L(G00 ), if G00 is the DRCG derived by the rule schemata 1, 2 and 300 .

*4 Maximal Use of Selectors*

*The (potential) adequacy of internal CGs to NLP has been studied in [8],*

*where two variants of the relation )*

*K;in*

*are de ned. 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 de ne 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 rst variant a derivation is in the maximal local mode, and we write*

*) K;M l*

*for the corresponding derive relation, i x = x1 x2 x3 , y = x1 ux2 vx3, for*

*x 2 S , (u; v) 2 C , for some 1 i n, and there are no x0 , x0 , x0 2 T*

*2 i i*

*1 2 3*

*s.t. x = x0 x0 x0 , x0 2 S , and jx0 j jx j, jx0 j jx j, jx0 j > jx j. In that case,*

*1 2 3 2 i*

*1 1 3*

*3 2 2*

*the word-selector x2 is maximal in Si.*

*The translation of a CG K = (T; A; (S1; C1);::: ; (Sn; Cn)) with internal* derivations in maximal local mode into an equivalent DRCG only di ers 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 300 is changed into*

*3000 : [S ;C ](X uX vX ) ! [S ] (X uX vX ; uX v; X )*

*i i 1 2 3 i Ml 1 2 3 2 2*

*[Si](X2 ) catenate(S; X1; X2; X3 )*

*in which the ternary nonterminal [Si]Ml, 1 i n checks the maximal local*

*mode constraint. All predicates of the form [S ] (Y; X0 ;X ) are s.t. X*

*is a*

*i Ml 2 2 2*

*subrange of X0 , and X0 is itself a subrange of Y . Let x*

*be the string selected*

*2 2 2*

*by X*

*and let ux v be the string selected by X0*

*for some u; v 2 T , and let*

*2 2 2*

*y = x x0 ux vx0 x be the string selected by Y for some x ; x0 ; x0 ;x*

*2 T .*

*1 1 2 3 3 1 1 3 3*

*In that case, the predicate call [S ] (Y; X0 ;X ) is true i we have x0 x0*

*6= "*

*i Ml 2 2 1 3*

*and x0 x x0 2 S , that is the word-selector x is not maximal in S , and the*

*1 2 3 i 2 i*

*negative call [S ] (Y; X0 ;X ) succeeds i the word-selector x*

*is maximal in*

*i Ml 2 2 2*

*Si. Thus, 8i 2 f1;::: ; ng, each [Si]Ml can be de ned by the clause schema*

*4 : [S ] (X X0 X0 X0 X ; X0 ;X ) ! null (X0 ; X0 )*

*i Ml*

*1 1 2 3 3 2 2*

*2 1 3*

*catenate([S ]; X0 ;X ; X0 )*

*i 1 2 3*

*The binary nonterminal null2 is true i its arguments denote empty ranges* null2("; ") ! "

*4.2 Maximal Global Mode*

*In this second varianta derivation is in the maximal global mode, and we write*

*) K;M g*

*for the corresponding derive relation, i x = x1 x2 x3 , y = x1 ux2 vx3, for*

*x 2 S , (u; v) 2 C , for some 1 i n, and there are no x0 , x0 , x0 2 V*

*2 i i*

*1 2 3*

*s.t. x = x0 x0 x0 , x0 2 S , for some 1 j n, and x0*

*jx j, jx0 j jx j,*

*1 2 3 2 j*

*1 1 3 3*

*jx0 j > jx j. In that case, the word-selector x is maximal w.r.t. all selectors*

*2 2 2*

*S1;::: ; Sn.*

*For the translation of an internal CG K = (T; A; (S1; C1);::: ; (Sn; Cn)) in maximal global mode into an equivalent DRCG, the clause schema 3000 is*

*changed into*

*30000 : [S ;C ](X uX vX ) ! Mg(X uX vX ; uX v; X )*

*i i 1 2 3*

*1 2 3 2 2*

*[Si](X2) catenate(S; X1; X2; X3)*

*where the nonterminal Mg de nes the maximal global mode constraint. For* each i 2 f1;::: ; ng, this constraint can be de ned by the following Mg-clauses

*Mg(X1; X2; X3) ! [Si]Ml(X1; X2; X3)*

*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 = fanbncn j n 1g, is a CL that can be de ned by an internal CG K = (fa; b; cg; fabcg; (b+; f(a; bc)g), in maximal local or maximal global mode. Since in K there is only a single selector language, the local and global max- imal modes are identical. When applied to K, the transformation schema of Section 4.1 gives an equivalent DRCG G whose clauses are

*S(abc) ! "*

*S(X) ! [b+; f(a; bc)g](X)*

*[b+; f(a; bc)g](X aX bcX ) ! [b+] (X aX bcX ; aX bc; X )*

*1 2 3*

*Ml 1 2 3 2 2*

*[b+](X ) catenate(S; X ;X ;X )*

*[b+]*

*2*

*(X X0 X0 X0 X ; X0 ;X ) ! null (X0 ; X0 )*

*1 2 3*

*Ml 1 1*

*2 3 3 2 2*

*2 1 3*

*catenate([b+]; X0 ;X ; X0 )*

*1 2 3*

*null2("; ") ! "*

*[b+](b) ! "*

*[b+](bX) ! [b+](X)*

*Note that the last two clauses de ne the selector language b+.*

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

*Using K, we have abc )*

*K;M l*

*aabbcc )*

*K;M l*

*aaabbbccc in which at each step the*

*word-selector has been underlined.*

*If we consider DRCG derivations, starting from the instantiation S(h0::9iw)* of the axiom S on w, we can build the complete derivation

*2*

*) [b+ G*

*3000*

*; f(a; bc)g](h0::9iw)*

*) [b+]Ml(h0::9iw; h2::7iw; h3::5iw)*

*G*

*[b+](h3::5iw) catenate(S; h0::2iw; h3::5iw; h7::9iw)*

*+*

*) S(h0::6iw0 =aabbcc) G*

*2*

*) [b+ G*

*3000*

*; f(a; bc)g](h0::6iw0 )*

*) [b+]Ml(h0::6iw0 ; h1::5iw0 ; h2::3iw0 ) G*

*[b ](h2::3iw0 ) catenate(S; h0::1iw0 ; h2::3iw0 ; h5::6iw0 )*

*+*

*+*

*) S(h0::3iw00 =abc) G*

*1*

*) "*

*G*

*which shows that w 2 L(G). 8*

*5 Other Local Variants*

*Two local variants of internal CG, de ned 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 deriva- tion can de ne 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 rst local variant, for any CG K = (T; A; (S1; C1);::: ; (Sn; Cn)),*

*Ilie de nes a local derive relation, denoted ) , as follows.*

*If ) K;in*

*K;in;loc*

*is the usual internal derive relation of Section 3 and if we consider*

*two strings z; x 2 T , s.t. z )*

*x, z = z z z , z 2 S , (u; v) 2 C , for some*

*K;in*

*1 2 3 2 i i*

*1 i n, and x = z1uz2vz3, we have x )*

*K;in;loc*

*y, w.r.t. z )*

*K;in*

*x, and we write*

*z ) K;in*

*x )*

*K;in;loc*

*y, i we have u = u1u2, v = v1v2, x = x1x2x3 for x1 = z1u1,*

*x2 = u2z2v1, x3 = v2z3, x2 2 Sj, (w; s) 2 Cj, for some 1 j n, and*

*y = x1wx2sx3.*

*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 )*

*K;in;loc*

*called maximal local and*

*denoted ) . We have z ) x )*

*y i z ) x )*

*y and there is*

*K;in;M loc*

*K;in*

*K;in;M loc*

*K;in K;in;loc*

*no other local derivation z )*

*K;in*

*x )*

*K;in;loc*

*y0 s.t. the decomposition used for x,*

*say x = x0 x0 x0 , x0 2 S , veri es jx0 j jx j, jx0 j > jx j and jx0 j jx j.*

*1 2 3 2 j*

*1 1 2 2 3 3*

*i*

*8*

*0*

*) means that this derivation step uses the clause schema i. Moreover, the reader is*

*G*

*invited to check that, for example, the third derivation step results from the fact that we*

*have both ([b+]*

*Ml (h0::9iw*

*; h2::7iw*

*; h3::5iw*

*); ")*

*+ 62)*

*G*

*and [b+](h3::5iw*

*+*

*) ) ".*

*G*

*For 2 floc; M locg and x; y 2 T , we write x*

*)*

*K;in;*

*y i we have a nite*

*derivation, in which each step, excepting the rst one, is performed in mode*

*w.r.t. the previous one, that is*

*x = x0 )*

*K;in*

*x1 ) K;in;*

*x2 ) K;in;*

*::: )*

*K;in;*

*xk = y*

*for some k 0, and, in the case = Mloc, the rst step x0 )*

*K;in*

*x1 must be*

*performed in the maximal mode.*

*5.1 From Local Variant Internal CG to RCG*

*The translation of the CG K = (T; A; (S1; C1);::: ; (Sn; Cn)) used with the*

*derive relation ) , into an equivalent RCG can be performed as follows.*

*K;in;loc*

*First, 8u 2 A, we specify that each axiom is a sentence by*

*S(u) ! "*

*Second, for each Si, 1 i n, and for each axiom u = u1u2u3 2 A,*

*u ;u ;u 2 T , we statically check if u is an element of S . If it is the case,*

*1 2 3 2 i*

*we generate a clause of the form*

*5 : S(u1XU2Y u3) ! x(u2; U2) [Si; Ci](XU2Y; U2 )*

*which speci es that the initial derivation step must be performed by ) . The*

*K;in*

*predicate name x, de ned by*

*x(X; X) ! "*

*is used to \anchor" a terminal string (here the occurrence of u2, between X* and Y in clause 5) at a given position by means of a variable (here U2 ).

*Third, for each i, 1 i n and for each (u; v) 2 Ci s.t. u1u2 = u; v1v2 = v, we generate*

*6 : [Si; Ci](u1Xu2Zv1Y v2 ; Z) ! [Si](Z) [S; C](Xu2Zv1 Y; u2Zv1 )*

*which processes the derivation steps in loc mode.*

*Finally, the predicate name [S; C] is de ned by the clauses*

*[S; C](X; Z) ! [Si; Ci](X; Z) 8i 2 f1;::: ; ng [S; C](Z; Z) ! "*

*which indicate that each derivation step, excepting the rst one, is performed*

*by a pair (Si; Ci), until completion.*

*Let us assume that the selectors Si of K are RCLs, and let L be the*

*language de ned by K with ) . We can show that the previous PRCG*

*K;in;loc*

*also de nes L and, moreover, that its sentences can be parsed in time O(l6)* (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 O(ld), L can be parsed in time* O(lmax(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 O(l6) time by an equivalent PRCG. This shows that the languages de ned by this local variant of CGs can be parsed in O(l6) 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; (S1; C1);::: ; (Sn; Cn)) used with*

*the derive relation ) , into an equivalent RCG is similar to the previous*

*K;in;M loc*

*one, except on two points. First the clauses of the form 5 are generated only*

*if they ful ll the maximal mode condition. Second, the clause schema 6 is* changed into

*[Si; Ci](U1Xu2Zv1Y V2; Z) ! x(u1; U1) x(v2; V2)*

*[Si](Z) [S; C](Xu2Zv1Y; u2Zv1)*

*Mloc(U1Xu2Zv1Y V2; Z; U1; u2Zv1; V2)*

*The nonterminal Mloc checks that the maximal mode condition is ful lled. 9 For each i, 1 i n and for each (u; v) 2 Ci s.t. u1u2 = u; v1v2 = v, we generate*

*Mloc(u1Xu2Zv1Y v2; Z; u1X1; X2; X3v2) ! [Si; Ci](u1Xu2Zv1Y v2; Z)*

*gtr(u2Zv1; X2)*

*The maximal parse time complexity is reached for the Mloc-clauses of* degree eight. This shows that the languages de ned by this maximal local variant of CGs can be parsed in O(l8) 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 de ned DRCGs. With this new formalism we have shown how CLs can be de ned, 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 uni ed* 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 de ne the selector languages. At least for selector languages which are [D]RCLs, the translation of CGs into equivalent [D]RCGs leads to a single de nition 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. Barth elemy, P. Boullier, Ph. Deschamp, and E . 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. P aun, 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. Mart n-Vide, and Gh. P aun. 1998. Contextual grammars as generative models of natural languages. Computational Linguistics, 24(2):245{ 274.*

*[9] S. Marcus. 1969. Contextual grammars. Revue Roumaine des Math ematiques Pures et Appliqu ees, 14(10):1525{1534.*

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