TALN 2004, Fès, 19–21 avril 2004

Les Grammaires à Concaténation d'Intervalles (RCG) comme formalisme grammatical pour la linguistique

Benoît Sagot (1,2) et Pierre Boullier (1)
(1) Projet ATOLL - INRIA
Domaine de Voluceau, B.P. 105, 78153 Le Chesnay Cedex, France
{benoit.sagot;pierre.boullier}@inria.fr
(2) TALaNa/Lattice - Université Paris 7
2 place Jussieu, 75251 Paris Cedex 05, France

Résumé - Abstract

Le but de cet article est de montrer pourquoi les Grammaires à Concaténation d'Intervalles (Range Concatenation Grammars, ou RCG) sont un formalisme particulièrement bien adapté à la description du langage naturel. Nous expliquons d'abord que la puissance nécessaire pour décrire le langage naturel est celle de *PTIME*. Ensuite, parmi les formalismes grammaticaux ayant cette puissance d'expression, nous justifions le choix des RCG. Enfin, après un aperçu de leur définition et de leurs propriétés, nous montrons comment leur utilisation comme grammaires linguistiques permet de traiter des phénomènes syntagmatiques complexes, de réaliser simultanément l'analyse syntaxique et la vérification des diverses contraintes (morphosyntaxiques, sémantique lexicale), et de construire dynamiquement des grammaires linguistiques modulaires.

The aim of this paper is to show why Range Concatenation Grammars (RCG) are a formalism particularly suitable to describe natural language. We first explain that the power necessary to describe natural language is that of *PTIME*. Then, among grammatical formalisms that have this expressing power, we justify the choice of RCGs. Finally, after an overview of their definition and properties, we show how their use as linguistic grammars makes it possible to deal with complex syntactic phenomena, to achieve simultaneously both syntactic parsing and constraints checking (e.g., morphosyntactic and/or lexical semantic constraints), and to build dynamically modular linguistic grammars.

Mots-clefs – Keywords

Grammaires de réécriture, Grammaires Faiblement Contextuelles, complexité du langage naturel, Grammaires à Concaténation d'Intervalles (RCG)
Rewriting Systems, Mildly Context-Sensitive Grammars, Complexity of Natural Language, Range Concatenation Grammars (RCG)

1 La complexité du langage naturel

1.1 Les grammaires faiblement contextuelles (MCS)

La puissance formelle permettant de définir la syntaxe du langage naturel fait encore l'objet de discussions. Il a été démontré par (Schieber, 1985) que les Grammaires Non-Contextuelles (CFG) sont insuffisantes pour décrire certains phénomènes linguistiques. De ce fait, (Joshi, 1985) a introduit une classe de grammaires à la puissance formelle légèrement supérieure à celle des CFG : il s'agit des grammaires faiblement contextuelles (mildly context-sensitive, ou MCS). Encore aujourd'hui, on considère souvent que les grammaires MCS ont la puissance adéquate pour décrire le langage naturel. Elles sont définies par les propriétés suivantes :

MCS1. les langages MCS contiennent strictement les Langages Non-Contextuels,

MCS2. les langages MCS peuvent être analysés en temps polynomial,

MCS3. les langages MCS doivent reconnaître certains langages à dépendances complexes, comme $\{a^n b^n c^n\}$, $\{a^n b^m c^n d^m\}$ ou le langage 2-copy,

MCS4. les langages MCS vérifient la Propriété de Croissance Constante (ci-après CGP)[1].

Les grammaires MCS proposées aujourd'hui pour l'analyse linguistique sont d'une part les Grammaires d'Arbres Adjoints (TAG) et les grammaires qui lui sont faiblement équivalentes, et d'autre part, avec un pouvoir d'expression plus grand, les Systèmes de Réécriture Linéaire Non-Contextuels (LCFRS) et ses divers équivalents[2]. À notre connaissance, aucun formalisme grammatical MCS n'a été proposé qui soit strictement plus puissant que ceux qui rentrent dans le cadre des LCFRS, bien qu'il existe des langages se rangeant strictement dans cet intervalle.

1.2 Nombres chinois et génitifs en géorgien ancien

Il existe dans certaines langues des constructions sporadiques qui ne sont pas analysables par des grammaires MCS car elles ne respectent pas la CGP (condition MCS4). Classiquement, il en est ainsi des nombres chinois (Radzinski, 1991) et du génitif en géorgien ancien[3] (Michaelis et Kracht, 1996). En revanche, les Grammaires Non-Contextuelles Parallèles Multiples (PMCFG) introduites par (Kaji et al., 1992) couvrent ces deux cas tout en étant strictement moins puissantes que les formalismes couvrant tout *PTIME* (Groenink, 1996).

Naturellement, on pourrait considérer ces cas comme des cas marginaux dont il n'est pas nécessairement souhaitable de pouvoir rendre compte. En particulier, dans le cas des nombres chinois, il peut être objecté (Radzinski, 1991) que les contraintes qui font que la CGP n'est pas respectée pourraient ne pas être de nature linguistique, mais être issues des propriétés mathématiques des objets qu'il dénote. Par ailleurs, il serait satisfaisant de disposer d'une grammaire pour le géorgien ancien permettant un nombre arbitraire de génitifs empilés, mais il n'est pas toujours considéré que cela soit indispensable (Kallmeyer, 1997). Cependant, au moins deux autres phénomènes, bien moins sporadiques, mettent en cause l'idée selon laquelle les langages MCS suffisent au traitement des langues. Il s'agit des exemples bien connus que sont le «scrambling» et les constructions coordonnées à verbes multiples.

[1] Un langage vérifie la CGP (pour Constant Growth Property) si, lorsque l'on ordonne par taille croissante les chaînes du langage, on peut majorer la différence entre les longueurs de deux chaînes consécutives.

[2] Comme les TAG Multi-Composants (MC-TAG) ou les Grammaires Minimalistes (MG)

[3] Les auteurs cités montrent que le génitif en géorgien ancien n'est pas semi-linéaire. En réalité le génitif en géorgien ancien, pouvant être réduit à $\{a_1 b\ a_2 g b\ \ldots a_k g^{k-1} b\}$, ne respecte même pas la CGP, ce qui est plus fort.

1.3 Scrambling en allemand et coordonnées à verbes multiples en néerlandais

Le scrambling en allemand[4] peut être décrit de la façon suivante : dans une subordonnée comprenant des complétives, les compléments des différents verbes sont placés entre le sujet du verbe principal et les verbes. Les verbes sont alors généralement ordonnés du plus enchâssé au verbe principal, mais les compléments sont dans un ordre quelconque (on trouvera des exemples dans (Becker et al., 1991)). Il est montré par (Boullier, 1999b) que le langage utilisé par (Becker et al., 1992) pour modéliser le scrambling est insuffisant, et utilise un langage plus vaste qu'il appelle SCR[5] Ce langage est au-delà de la puissance des TAG (Becker et al., 1991) et même de celle des LCFRS (Becker et al., 1992). Il semble pourtant vérifier la CGP. Cependant, comme indiqué plus haut, aucun formalisme grammatical n'a été proposé jusqu'à présent qui soit simultanément MCS tout en ayant une puissance strictement plus grande que les LCFRS. Il semble donc raisonnable de chercher un formalisme grammatical qui ne soit pas MCS pour décrire le scrambling en allemand[6].

Les constructions coordonnées à verbes multiples en néerlandais sont analysées par (Groenink, 1996) : il exhibe une classe d'exemples C_1 non exprimables dans le cadre des LCFRS[7], qui regroupe des exemples ayant un nombre de verbes arbitraire, ce qui nécessite une argumentation sur la pertinence de cet argument (cf. ci-dessous). Il a été prouvé par la suite qu'elle n'était pas même semi-linéaire (Michaelis et Kracht, 1996). Cependant, bien qu'une PMCFG puisse analyser cette classe d'exemples, (Groenink, 1996) montre qu'on peut l'étendre à une classe plus large C_2, encore moins acceptable linguistiquement que C_1, que même les PMCFG, qui subsument strictement les LCFRS et ne respectent pas la CGP, ne peuvent décrire[8].

1.4 La puissance nécessaire pour décrire le langage naturel est *PTIME*

Même en rejetant la classe C_2 d'exemples du néerlandais, les exemples ci-dessus semblent indiquer que les formalismes MCS ne suffisent pas à décrire le langage naturel[9]. Toutefois, les trois premières conditions MCS semblent devoir être vérifiées par les langues naturelles : MCS1 a été justifiée plus haut, MCS3 a été explicitement introduite pour que les grammaires MCS puissent traiter divers phénomènes linguistiques courants, et MCS2 est justifiée au moins par les besoins pratiques de complexité. Seule MCS4 peut donc être violée : la modélisation du langage naturel passe par des grammaires ne vérifiant pas la CGP.

---

[4]Ce phénomène existe dans d'autres langues, comme le japonais.

[5]Il s'agit approximativement du langage suivant : soit un alphabet de terminaux partitionné en $\{n_1, \ldots, n_l\}$ et $\{v_1, \ldots, v_m\}$ et muni d'une correspondance h indiquant, si $v = h(n)$, que n est un argument de v. Une suite de terminaux est reconnue par SCR si et seulement si elle est de la forme $\pi(n_1, \ldots, n_p)v_1 \ldots v_q$ où π est une permutation, les n_i et les v_j sont deux à deux distincts, et pour tout n_i il existe un et un seul v_j tel que $h(n_i) = v_j$.

[6]On peut noter dès ici que le fait qu'un tel formalisme ne serait peut-être pas de la puissance minimale pour exprimer SCR n'est pas nécessairement un problème. En effet, si l'on peut exhiber des grammaires analysables sans véritable surcoût par rapport aux grammaires MCS mais permettant d'exprimer simplement et élégamment le scrambling, rien ne justifie dans la pratique la quête de la plus petite grammaire possible incluant SCR.

[7]Il s'agit de son fragment (1.14) répété en (3.39).

[8]Il s'agit du cas où l'on ne se restreint pas à un seul verbe à montée à l'infinitif parmi les verbes multiples coordonnés (cf. paragraphe 3.3 de (Groenink, 1996)).

[9]Notons à cet égard que si l'on accepte la classe d'exemples C_2, l'insuffisance des PMCFG à les traiter ne prouve rien de plus : on pourrait imaginer trouver un formalisme grammatical agréable subsumant les grammaires MCS mais non comparable avec les PMCFG.

Toutefois, la problématique de la distinction compétence/performance ne rend pas la chose si simple, par exemple pour la classe d'exemples C_1 du néerlandais. De fait, (Manaster-Ramer, 1987) indique que le résultat dépend du point de vue que l'on adopte sur ce que l'on inclut dans la langue et ce que l'on en exclut. Toutefois, et parallèlement à la note 6 du paragraphe 1.3, il nous semble artificiel de vouloir chercher la grammaire minimale couvrant une langue donnée, en allant jusqu'à s'autoriser à borner *ex abrupto* certains paramètres sans justification linguistique claire, surtout s'il existe des classes de grammaires permettant de s'affranchir de telles limitations sans pour autant entraîner un surcoût significatif.

La problématique compétence/performance est encore plus forte concernant l'insuffisance des PMCFG pour décrire la classe C_2. En réalité, on peut penser qu'une telle classe d'exemples, peu convaincante en termes de performance mais acceptable (quoiqu'à la marge) en termes de compétence, est plus ou moins du niveau maximal de complexité de la langue concernée, lequel est disponible mais presque jamais utilisé dans une grammaire de la langue. Ainsi, un formalisme grammatical permettant de traiter le néerlandais devrait avoir la puissance nécessaire au traitement de C_2. Mais il devrait pour ce faire exprimer sa pleine puissance, utilisée très rarement dans une grammaire complète du néerlandais. En poursuivant ce raisonnement, et puisque les PMCFG ne suffisent pas, on ne peut que conclure que le niveau de complexité requis est le seul autre niveau immédiatement supérieur à celui des grammaires MCS pour lequel il existe des formalismes grammaticaux aux bonnes propriétés, à savoir tout *PTIME* (cf. partie 2).

Par ailleurs, nous n'avons parlé jusqu'à présent que de puissance générative faible. Or si l'équivalence faible entre le néerlandais et un langage analysable avec une PMCFG n'est pas certaine, l'«équivalence forte» entre structures linguistiques et analyse par une PMCFG, que l'on souhaite évidemment, est encore plus douteuse. Cela renforce l'intuition selon laquelle les grammaires adéquates à l'analyse des langues ne respectent pas la CGP, ne sont pas même exprimables par une PMCFG, et nécessitent un formalisme grammatical couvrant tout *PTIME*.

2 Les Grammaires à Concaténation d'Intervalles (RCG)

2.1 Pourquoi les RCG

Les classes de grammaires proposées dans la littérature ne vérifiant pas la CGP[10] mais satisfaisant les trois autres conditions MCS1, MCS2 et MCS3 ne sont pas en nombre très important[11]. Dans la plupart des cas, il s'agit de classes de grammaires couvrant exactement *PTIME*. Certains sont moins puissants, comme les PMCFG, mais on a vu leur insuffisance dans la partie 1. D'autres formalismes non-MCS ont été proposés qui ne sont pas analysables en temps polyno-

[10] Les véritables conséquences de la violation de CGP sur la nature d'une classe de grammaire sont peu claires. Toutefois, d'une part pour des raisons intuitives et d'autre part à travers les classes de grammaires proposées à ce jour et ne respectant pas CGP, il semble qu'il y ait un lien entre le respect de CGP et le fait d'être «linéaire» au sens des LMG ou des RCG, c'est-à-dire le fait que deux non-terminaux de l'analyse ne peuvent avoir dans leurs «portées» respectives, quoi que cela puisse dire dans un formalisme donné, un terminal en commun : le non-respect de CGP passerait par une certaine forme de non-linéarité se traduisant par l'utilisation multiple de mêmes terminaux au cours de l'analyse. Il en est ainsi des PMCFG, des LMG, des RCG, et des MC-TAGS à nœuds partagés (restreintes ou non) proposés par (Kallmeyer, 2003). À vrai dire, la condition CGP doit peut-être être remplacée dans ces considérations par celle de semi-linéarité. Mais dans tous les cas, si ces réflexions s'avèrent justifiées, elles indiqueraient la nécessité de grammaires non-linéaires pour traiter le langage naturel.

[11] Il est à noter que la condition MCS3, qui est imposée par divers phénomènes linguistiques, rend inadéquats des formalismes comme les V-TAGs (Rambow, 1994) ou les D-Tree Substitution Grammars (Rambow et al., 2001).

mial, qu'ils subsument *PTIME* ou non, mais ils ne sont pas satisfaisants en termes de complexité (condition MCS2). Parmi les formalismes couvrant exactement *PTIME*, on trouve en particulier la variante dite «simple» des Litteral Movement Grammars (sLMG) introduites par (Groenink, 1996) et les Grammaires à Concaténation d'Intervalles (Range Concatenation Grammars, ci-après RCGs) introduites par (Boullier, 2000a). Tout en restant polynomiales (MCS2), elles peuvent traiter les cas présentés en partie 1 (Groenink, 1996; Boullier, 1999a; Boullier, 1999b).

Parmi ces deux formalismes, les RCG définissent des prédicats sur des intervalles de la chaîne d'entrée, alors que les sLMG considèrent des sous-chaînes. Or il semble plus naturel d'un point de vue linguistique de considérer non pas des sous-chaînes, mais plutôt des intervalles de la chaîne d'entrée, en particulier lorsqu'un même mot (i.e. un terminal) est présent plus d'une fois dans la phrase (i.e. la chaîne)[12]. C'est encore plus vrai si l'on a moins de terminaux que de mots de la langue, choix raisonnable mais en réalité non nécessaire moyennant des méthodes appropriées (cf. Partie 3). En outre, il est de plus en plus clair que la description du langage naturel nécessite la prise en compte d'informations de nature topologique (Gerdes et Kahane, 2001; Clément et al., 2002). Or il est facile d'encoder de telles informations dans une RCG qui manipule des intervalles, et donc des positions dans la chaîne, mais plus difficile dans une sLMG qui ne manipule que des sous-chaînes[13].

2.2 Les RCG : exemple et propriétés

Pour une définition formelle des RCG, leur complexité exacte d'analyse, et la démonstration de leurs propriétés, on se reportera par exemple à (Boullier, 2000a). Nous nous contenterons ici de présenter sur un cas particulier le fonctionnement des RCG.

Une RCG se présente sous la forme de *clauses*, qui ont la forme $P \rightarrow P_1 \dots P_n$, où P et les P_i sont des *prédicats* (P est le prédicat de partie gauche, les P_i sont ceux de partie droite). Une telle clause s'interprète ainsi : «si les prédicats P_1 à P_n sont vérifiés, alors le prédicat P l'est aussi». Les prédicats ont un ou plusieurs arguments. Chaque argument est la concaténation d'une ou plusieurs variables, chaque variable représentant un intervalle de la chaîne (et pas une sous-chaîne) ou un terminal (dont le nom commence par une minuscule ou un guillemet). La concaténation de plusieurs variables dénote donc aussi un intervalle, mais les intervalles concaténés doivent être successifs. Une clause comme $P(XY) \rightarrow P_1(X)\, P_2(XY)\, P_3(X, Y)$ peut donc se comprendre comme «le prédicat P appelé sur l'intervalle dénoté par XY de la chaîne d'entrée peut être remplacé par le prédicat P_1 appliqué à l'intervalle X, le prédicat P_2 à l'intervalle XY et le prédicat P_3 au couple (X, Y)» (on dit que les prédicats de partie droite sont *appelés*). Le prédicat de partie gauche de la première clause est l'axiome, et prend en entrée un intervalle couvrant toute la chaîne. Un prédicat est vérifié sur un intervalle si et seulement si l'on peut en dériver la chaîne vide à l'issue de remplacements successifs. Une chaîne est dans le langage si et seulement si le prédicat axiome est vérifié sur l'intervalle couvrant toute la chaîne.

[12]La différence fondamentale entre les RCG et les sLMG est bien celle qu'il y a entre traiter des intervalles et traiter des sous-chaînes. Or il est très facile de traiter en RCG toutes les occurrences d'une sous-chaîne qui se répète, puisqu'il suffit de considérer deux des intervalles concernés, de vérifier qu'ils correspondent à deux sous-chaînes égales, puis de traiter ces deux chaînes. En revanche, distinguer deux sous-chaînes égales correspondant à deux intervalles distincts est basique en RCG mais délicat en sLMG, puisque dans le pire des cas il faut utiliser la chaîne entière pour distinguer deux occurrences de la même sous-chaîne. Donc même indépendamment de considérations linguistiques, les RCG semblent plus facile d'emploi que les sLMG.

[13]L'utilisation des RCG pour écrire des grammaires topologiques est une direction de recherche très prometteuse, et fera probablement l'objet de publications ultérieures.

Nous illustrerons ceci par la RCG suivante (Boullier, 2003) qui reconnaît le langage 3-copy $\{www|w \in \{a,b\}^*\}$ (hors de portée des TAG), trivialement extensible aux langages k-copy :

$S(XYZ)$	$\rightarrow$	$A(X,Y,Z)$
$A(aX',aY',aZ')$	$\rightarrow$	$A(X',Y',Z')$
$A(bX',bY',bZ')$	$\rightarrow$	$A(X',Y',Z')$
$A(\varepsilon,\varepsilon,\varepsilon)$	$\rightarrow$	ε

La première clause a pour partie gauche l'axiome, qui prend donc comme argument un intervalle couvrant toute la chaîne. Il découpe cet intervalle en 3 sous-intervalles, qui sont nommés respectivement X, Y et Z, qui, si la chaîne est valide, seront les intervalles (distincts) correspondant aux trois sous-chaînes identiques. Puisque c'est la seule clause où S intervient, on est forcé de remplacer l'axiome par $A(X,Y,Z)$. À ce stade, il y a quatre possibilités :

– ou bien la deuxième clause peut s'appliquer, c'est-à-dire que chacun des trois intervalles commence par le terminal a, et alors on remplace $A(X,Y,Z)$, c'est-à-dire $A(aX',aY',aZ')$, par $A(X',Y',Z')$. On a ainsi «consommé» un a à l'initiale de chaque intervalle (ceci est un exemple du fait que l'analyse par une RCG n'est pas une analyse gauche-droite)[14] ;
– ou bien la troisième clause peut s'appliquer (même situation mais avec le terminal b) ;
– ou bien les trois intervalles sont vides (borne inférieure et borne supérieure sont égales), et on peut remplacer $A(\varepsilon,\varepsilon,\varepsilon)$ par ε, ce qui signifie que la chaîne initiale est dans le langage ;
– ou bien aucune clause ne s'applique, et ε ne peut être dérivé : la chaîne n'est pas reconnue.

Si l'on est dans l'un des deux premiers cas, on recommence jusqu'au succès ou à l'échec, en consommant en parallèle à chaque fois un terminal à l'initiale de chacun des trois intervalles, à condition qu'ils désignent la même chaîne. Si on réussit à réduire les intervalles à des intervalles vides, c'est que les terminaux des intervalles initiaux se correspondaient correctement.

Les RCG permettent d'appeler la négation d'un prédicat : on note par exemple $\overline{A(X,Y)}$ la négation de $A(X,Y)$). Un tel appel négatif est vérifié si et seulement si son appel positif échoue (Boullier, 2000a), ce qui implique d'interdire les grammaires inconsistantes, c'est-à-dire les grammaires qui peuvent générer des dérivations où un prédicat peut dériver sa propre négation.

Il est prouvé par (Boullier, 2000a) qu'une RCG peut être analysée en un temps polynomial, l'exposant dépendant de la grammaire (c'est la somme du nombre maximum d'arguments d'un prédicat et du nombre maximum de variables présentes dans une partie gauche). Par ailleurs, Boullier a développé un analyseur pour les RCG qui a précisément cette complexité.

Enfin, (Boullier, 2000a) montre que les RCG bénéficient de propriétés très intéressantes : sans modifier les grammaires en jeu, et par le seul ajout d'une à deux clauses, on peut construire à partir de deux RCG les grammaires définissant l'union des deux langages correspondants, leur concaténation, leur intersection, l'itération de Kleene de l'un d'eux, et le complément de l'un d'eux. Ceci peut être résumé, en reprenant la définition de la *modularité* donnée dans (Boullier, 2000a), en disant que les RCG sont modulaires par rapport à ces cinq opérateurs. De plus, la non-linéarité des RCG permet de créer des grammaires mêlant différents points de vue sur une même chaîne. Ces deux propriétés sont très intéressantes pour la linguistique. D'une part, la modularité permet d'imaginer des bibliothèques de modules linguistiques dans lesquelles on pourrait piocher pour construire des grammaires d'une langue, introduisant un à un divers phénomènes[15]. D'autre part, la non-linéarité permet l'écriture de grammaires apportant

[14] On notera que l'utilisation d'un nom de variable X' différent de X est inutile, mais facilite l'explication.

[15] Dans (Boullier, 2003), l'exemple du paradigme «règle générale avec exceptions» est explicité : pour construire une RCG reconnaissant un langage $\mathcal{L}$ défini à partir de trois autres langages $\mathcal{R}$, $\mathcal{S}$, $\mathcal{E}$ par $\mathcal{L} = \mathcal{R} \cap \overline{\mathcal{S}} \cup \mathcal{E}$, où $\mathcal{R}$ représente le langage où la règle générale est toujours vraie, $\mathcal{S}$ le sous-ensemble de $\mathcal{R}$ qui est à rejeter, et $\mathcal{E}$ le

simultanément divers points de vue sur la phrase (syntaxe, morphologie, sémantique lexicale). La partie suivante montre sur quelques exemples ce que permettent effectivement les RCG.

3 RCG pour la linguistique

3.1 Phénomènes syntaxiques complexes

Les phénomènes les plus difficiles rencontrés dans les langues, parmi lesquels ceux cités en partie 1, peuvent tous être analysés correctement dès lors que l'on dispose de la puissance des RCG, i.e. de tout *PTIME*. De fait, une RCG pour l'analyse des nombres chinois est proposée dans (Boullier, 1999b), de même qu'une RCG pour le «scrambling» en allemand tel que défini en 2.2. De plus, (Kallmeyer, 2003) montre que le formalisme qu'elle définit spécifiquement pour analyser correctement ce phénomène est tel qu'on peut facilement convertir une de ses grammaires en une RCG. Par ailleurs, le génitif en géorgien ancien est fondamentalement similaire aux nombres chinois, et peut être traité par une grammaire similaire. Enfin, les constructions coordonnées à verbes multiples en néerlandais peuvent être analysé adéquatement par une sLMG, comme le montre (Groenink, 1996), et donc également par une RCG aux analyses similaires.

En plus du côté théorique de la puissance d'expression des RCG, leur non-linéarité permet de disposer de différents points de vue sur une même chaîne ou sous-chaîne. Une première illustration de ce phénomène, en se restreignant à la syntaxe, est fourni par le phénomène des coordinations à ellipses. Soit par exemple la phrase : «Luc aime Marie et Jean Anne.» On peut la gloser par «Luc aime Marie et Jean aime Anne». Autrement dit, le verbe «aime» est «sous-entendu» dans le deuxième membre de phrase, ce qui peut être interprété comme le fait qu'il est utilisé deux fois[16]. La RCG suivante, fournit une analyse convenable de cette phrase :

$PCOORD(S_1\, V\, O_1\, "et"\, S_2\, O_2)$	$\rightarrow$	$P(S_1, V, O_1)\ P(S_2, V, O_2)\ V(V)$
%Une phrase se découpe en un sujet, un verbe et un objet qui doivent vérifier P, un "et", et		
%un autre sujet et un autre objet qui, avec le verbe, doivent vérifier P aussi		
$P(S, V, O)$	$\rightarrow$	$SUJ(S, V)\ OBJ(O, V)$
%Trois intervalles vérifient P si le premier est sujet du second, et le troisième objet du second		
$SUJ(S, V)$	$\rightarrow$	$GN(S)\ ACC\_G\_N(S, V)$
%Le sujet d'un verbe est un groupe nominal et s'accorde en genre et en nombre avec lui		
$OBJ(O, V)$	$\rightarrow$	$GN(O)$ %L'objet d'un verbe est un groupe nominal
$GN(X)$	$\rightarrow$	$NP(X)$ %Un nom propre est un groupe nominal
$ACC\_G\_N("Luc", "aime")$	$\rightarrow$	ε %"Luc" est accordé en genre et nombre avec "aime"
$ACC\_G\_N("Jean", "aime")$	$\rightarrow$	ε %"Jean" est accordé en genre et nombre avec "aime"
$V(aime)$	$\rightarrow$	ε %"aime" est un verbe
$NP("Luc")$	$\rightarrow$	ε %"Luc" est un nom propre
$NP("Marie")$	$\rightarrow$	ε %"Marie" est un nom propre
$NP("Jean")$	$\rightarrow$	ε %"Jean" est un nom propre
$NP("Anne")$	$\rightarrow$	ε %"Anne" est un nom propre

langage des exceptions par lequel il faut le remplacer, il suffit de rajouter les clauses $L(X) \rightarrow R(X)\ \overline{S(X)}$ et $L(X) \rightarrow E(X)$, où l'on a nommé l'axiome d'un langage par la lettre majuscule droite correspondant à son nom.

[16] L'analyse linguistique de ce phénomène est bien plus complexe, car d'une part d'autres constituants que le verbe peuvent être élidés (mais pas tous), et d'autre part le verbe «sous-entendu» n'a que certaines des propriétés du verbe plein (ainsi la personne dans «Je mange une pomme et Jean une poire»). Mais cet exemple est là pour montrer comment peut être utilisée la non-linéarité des RCG pour la description de la syntaxe, et non comment on doit analyser en RCG les coordinations à ellipse de manière linguistiquement cohérente.

Cette grammaire regroupe donc des informations que d'autres formalismes rejettent au-delà des grammaires de réécriture qui leur sont sous-jacentes. C'est le cas de la nature des relations syntaxiques (sujet, objet) et de l'accord (même si par souci de simplification ce dernier est traité de façon très rudimentaire). C'est tout l'intérêt des RCG que de pouvoir dire plusieurs choses sur le même intervalle de la chaîne, contrairement à d'autres formalismes (comme les TAG ou les LFG), qui sont contraints de ne conserver qu'une seule de ces informations, celle qui est la plus structurante (le plus souvent les relations de constituance), et de rejeter dans des décorations (traits) les autres informations (nature de la relation, accord, etc.). L'inconvénient majeur d'une telle approche est que l'analyse perd alors dans le cas général les bonnes propriétés qu'avait la grammaire de réécriture sous-jacente (une CFG en LFG, par exemple) et peut devenir exponentielle. De plus, dans la plupart des cas, l'analyse se fait d'abord sans tenir compte des décorations, générant un très grand nombre d'analyses possibles, qui sont ensuite filtrées à l'aide des traits. À l'inverse, les RCG peuvent traiter toutes les informations simultanément, rejetant dès le début les analyses non valides quelle qu'en soit la raison.

3.2 Interface syntaxe-sémantique intégrée à la grammaire

La non-linéarité des RCG permet non seulement l'utilisation multiple d'un même intervalle de la chaîne pour exprimer des relations syntaxiques, mais aussi d'avoir des points de vue véritablement différents sur des intervalles, et en particulier pour combiner la syntaxe et la sémantique (on se restreint ici à la sémantique lexicale, c'est-à-dire en gros aux cadres de sous-catégorisation et aux restrictions de sélection). C'est une généralisation de la remarque finale du paragraphe précédent : non seulement on peut écrire en RCG une grammaire gérant toute la syntaxe, y compris (entre autres) la nature des relations syntaxiques et les règles d'accord, mais en plus on peut faire en sorte que la RCG elle-même vérifie au cours de l'analyse les contraintes de sous-catégorisation et de restriction de sélection. Il suffit pour cela de rajouter des prédicats traitant des relations de sémantique lexicale, assortis de prédicats plaçant les terminaux dans une hiérarchie de types. Les interactions entre syntaxe et sémantique lexicale, comme la nature des syntagmes réalisant l'agent et le patient d'un verbe transitif selon le mode (actif ou passif) sont alors facilement représentables. Nous avons écrit et vérifié des grammaires jouets sur ce principe. C'est un progrès sur les formalismes qui n'utilisent ces contraintes que pour filtrer des analyses souvent nombreuses fournies par une première étape purement syntaxique.

3.3 RCG linguistiques modulaires dynamiques

La grammaire présentée en 3.1 fait le choix d'associer à chaque entrée lexicale un symbole terminal. Cependant, ce choix mène à des grammaires excessivement volumineuses, à la fois en termes de nombre de terminaux (au moins un par forme du lexique) et en termes de nombre de clauses (plusieurs clauses pour chaque forme, ne serait-ce que pour traiter ses caractéristiques morphosyntaxiques et de sémantique lexicale). C'est là que les propriétés de clôture des RCG interviennent. En effet, elles permettent d'associer à chaque forme une pseudo-grammaire (une grammaire sans axiome) qui contient toutes les propriétés qui lui sont associées, puis, pour une phrase donnée, de constituer dynamiquement la grammaire qui pourra l'analyser en regroupant sans les modifier les pseudo-grammaires associées à chaque mot de la phrase[17]

[17]Plus précisément, à chaque mot susceptible d'être dans la phrase, pour traiter les homonymes et d'autres phénomènes comme les locutions ou les mots composés.

avec une grammaire-noyau. De plus, rien n'interdit que ces pseudo-grammaires, ainsi que la grammaire-noyau, soient pré-compilées statiquement puis «linkées» dynamiquement pour construire très efficacement à la volée des grammaires locales d'analyse, par exemple phrase par phrase. De fait, nous avons implémenté un tel mécanisme pour générer des grammaires avec une grammaire-noyau réduite et un lexique-jouet[18].

L'analyseur développé par Boullier permet de compiler des RCG d'une façon qui rende l'analyse très efficace. De plus, il permet l'appel de prédicats externes au sein des RCG. Il est donc possible dans ce contexte de faire appel pour vérifier certains prédicats à d'autres mécanismes que des clauses RCG. Naturellement, ceci n'a d'intérêt en termes de complexité que si l'on se limite à des mécanismes exploitables en temps polynomial, et qui dont seraient exprimables à l'aide de RCG (il en est ainsi par exemple de certaines Logiques de Description). On peut envisager de disposer d'une ontologie (ou une base de connaissances) pour vérifier certains prédicats de sémantique lexicale (du type $CONCRET(X) \rightarrow COMESTIBLE(X)$), voire pour construire dynamiquement les pseudo-grammaires associées à chaque mot à l'aide de clauses sous-spécifiées et de propriétés héritées au sein de l'ontologie.

Pour la construction d'une grammaire «en vrai grandeur», les propriétés des RCG sont à nouveau très intéressantes. En effet, un volume de travail considérable a été fourni depuis de nombreuses années pour le développement de grammaires à large couverture dans d'autres formalismes. Or (Boullier, 1999a; Boullier, 2000b) montrent que l'on peut convertir le «squelette syntaxique» (sans les traits) de ces grammaires en des RCG équivalentes manipulant les mêmes concepts linguistiques sous-jacents. Dans le cas général, la conversion automatique des contraintes sur les traits (équations fonctionnelles en LFG, par exemple) est de plus tout à fait envisageable, si elles sont de complexité polynomiale. Enfin, comme indiqué plus haut, les RCG obtenues sont analysables avec la même complexité qu'avec les analyseurs dédiés des formalismes sources. Tout ceci autorise le développement de la grammaire-noyau non pas à partir de zéro, mais à partir de la conversion en une RCG RCG_{base} d'une grammaire existante G_{base}, et en la complétant (on peut penser par exemple à une TAG ou à une MCTAG). Cette complétion peut se faire soit en modifiant la grammaire convertie RCG_{base}, soit, profitant ainsi de la clôture des RCG par intersection, en écrivant un axiome appelant d'une part l'axiome de RCG_{base} et d'autre part, indépendamment, celui d'une RCG complémentaire RCG_{compl} (qui pourrait appeler autant que nécessaire l'axiome de RCG_{base}). Cette deuxième solution, plus simple dans un premier temps, est cependant moins générale. Elle a l'avantage de permettre à la grammaire source G_{base} d'évoluer de son côté et d'être convertie régulièrement en une nouvelle version de RCG_{base}, sans que RCG_{compl} n'aie besoin d'être modifiée.

4 Conclusion

Un certain nombre de phénomènes linguistiques qui se rencontrent dans diverses langues indiquent que le pouvoir d'expression adéquat pour la description du langage naturel est celui couvrant *PTIME*. Parmi les formalismes satisfaisant ce critère, les Grammaires à Concaténation d'Intervalles (RCG) semblent bien les plus adaptées à une utilisation linguistique. De fait, nous avons suggéré comment quelques phénomènes, comme les contraintes topologiques, la coordi-

[18]On peut faire un parallèle entre ces pseudo-grammaires et les «supertags», et plus généralement avec toutes les grammaires lexicalisées (ainsi LFG ou LTAG), tout en notant que la différence fondamentale avec ce qui se passe dans ces formalismes est qu'ici ce ne sont pas des structures combinées par la grammaire ou des décorations ajoutées à ces structures qui sont associées aux éléments du lexique, mais bien des «morceaux de grammaire».

nation à ellipse, le «scrambling», ou l'interface entre syntaxe et sémantique lexicale, peuvent être traités de façon appropriée avec les RCG. De plus, les propriétés des RCG permettent l'écriture de grammaires modulaires, ouvrant ainsi la voie à des grammaires dynamiques, issues d'une grammaire noyau et de pseudo-grammaires associées aux éléments du lexique, et qui pourraient combiner entre autres les contraintes syntaxiques, morphologiques et de sémantique lexicale. Enfin, la conversion aisée et efficace de TAG existantes permet d'envisager le développement à relativement court terme de RCG linguistiques opérationnelles.

Références

BECKER T., JOSHI A., RAMBOW O. (1991), Long-distance scrambling and Tree Adjoining Grammars, Actes de *EACL-91*, 21-26.

BECKER T., RAMBOW O., NIV M. (1992), The derivational generative power of formal systems, or scrambling is beyond LCFRS, *Technical Report IRCS-92-38*, University of Pennsylvania.

BOULLIER P. (1999a), On Multicomponent-TAG Parsing, Actes de *TALN-99*, 321-326.

BOULLIER P. (1999b), Chinese Numbers, MIX, Scrambling and Range Concatenation Grammars, Actes de *EACL-99*, 53-60.

BOULLIER P. (2000a), Range Concatenation Grammars, Actes de *IWPT 2000*, 53-64.

BOULLIER P. (2000b), On TAG parsing, *Traitement automatique des langues*, Vol. 41 (3), 761-793.

BOULLIER P. (2003), Counting with Range Concatenation Grammars, *Theoretical Computer Science*, Vol. 293, 391-416.

CLÉMENT L., GERDES K., KAHANE S. (2002), An LFG-type grammar for german based on the Topological Model, Actes de *LFG '02*, 116-129.

GERDES K., KAHANE S. (2001), Word order in German : A formal dependency grammar using a topological hierarchy, Actes de *ACL '01*, 116-129.

GROENINK A. (1996), Mild Context-Sensitivity and tuple-based generalizations of Context-Free Grammar, Actes de *MoL 4*, *Linguistics and Philosophy*, numéro spécial

JOSHI A. (1985), How much context-sensitivity is necessary for assigning structural descriptions : Tree Adjoining Grammars, *Natural Language Processing*, Cambridge University Press, New-York.

KAJI Y., NAKANISHI R., SEKI H., KASAMI T. (1992), The Universal Recognition Problems for Parallel Multiple Context-Free Grammars and for their Subclasses, *IEICE*, Vol. E75-D(4), 499-508.

KALLMEYER L. (1997), Local Tree Description Grammars, Actes de *MoL 5*, 77-84.

KALLMEYER L. (2003), Tree-local Multicomponent Tree Adjoining Grammars with Shared Nodes, manuscrit disponible à l'adresse `http://talana.linguist.jussieu.fr/~lkallmey/`.

MANASTER-RAMER A. (1987), Dutch as a formal language, *Linguistics and Philosophy*, Vol. 10, 221-246.

MICHAELIS J., KRACHT M. (1996), Semilinearity as a Syntactic Invariant, Actes de *LACL '96*, *Logical Aspects of Computational Linguistics*, Vol. 1328, 329-345.

RADZINSKI D. (1991), Chinese number-names, Tree Adjoining Grammars, and Mild Context-Sensitivity, *Computational Linguistics*, Vol. 17(3), 277-299.

RAMBOW O. (1994), Multiset-Valued Linear Index Grammars — Imposing Dominance Constraints on Derivations, Actes de *ACL '94*, 263-270.

RAMBOW O., VIJAY-SHANKER K., WEIR D. (2001), D-Tree Substitution Grammars, *Computational Linguistics*, Vol. 27(1), 87-121.

SCHIEBER S. (1985), Evidence against the context-freeness of natural language, *Linguistics and Philosophy*, Vol. 8, 333-343.