

Université de Bordeaux 1
Master Informatique, 2ème année
Bruno Mery

Initiation à la Recherche

Analysis of
*Proposal for a Natural Language Processing Syntactic
Backbone*
by Pierre Boullier

Sous la direction de Christian Rétoré.

Foreword

The automated processing of human-created languages (traditionnally – and quite inaccurately – called “Natural Languages” by computer scientists) is the motivation behind the creation of numerous computer and mathematical formalisms, now widely in use. The object of *Natural Language Processing* (NLP) is to generate a mean of describing the combination of the linguistic processes involved in human communication that can be processed by simple algorithms, and efficient enough for practical uses.

Context Free Grammars (CFGs), a simple formalism in favor for describing restricted languages (such as programming languages), have been proved un-sufficient for NLP. Since the research in this field began, various more powerful formalisms have been introduced, such as Tree Adjoining Grammars.

The contribution of Pierre Boullier towards this problem is a class of grammars that could be part of the answer. Range Concatenation Grammars (RCGs), based on an abstraction of Decisive Clause Grammars (such as those used in logical programming), can generate and parse the complete set of polynomial-time-parsable languages, which can be reasonably assumed to include human-understandable languages.

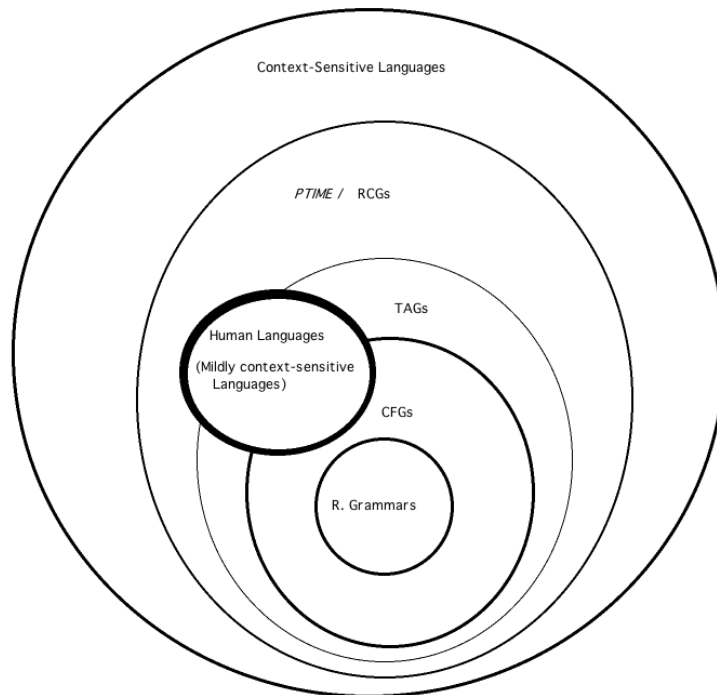


Figure 1: Place of RCG-parsable languages in Chomsky's Hierachy.

1 Introducing Range Concatenation grammars

In this section, we will study RCGs as they were introduced by Pierre Boullier, in his original articles [Bou98], [Bou99b] and [Bou99a].

1.1 Definitions

A RCG is a set of *rules* consisting of *predicates* over *ranges* of the sentence.

In [Bou99a], RCGs are formally defined, given a set of terminal symbols T ($\{a, b, c, \dots\}$) and a set of variable symbols V ($\{X, Y, Z, \dots\}$), as a set of *clauses*

$$\Psi_0 \rightarrow \Psi_1 \dots \Psi_n$$

defined over *predicates* Ψ_i of a given arity p :

$$\Psi_i = A(\alpha_1, \dots, \alpha_p)$$

where $\alpha_i \in (T \cup V)^*$; among them the *start predicate*, S . These predicates are used to define inductive proprieties over substrings of the sentence, in that akin to non-terminal symbols in a CFG clause.

Moreover, variable symbols may not appear more than once in a given predicate (ie., $A(X, X)$ is not valid). Also, for practical purposes, negative predicates can be used: $\overline{A(X)}$ is true when $A(X)$ is false. This does not add anything to the formal power of RCGs: the set of values satisfying a predicate, at any given moment in a particular analysis, is finite, and $\overline{A(X)}$ merely represents the (finite) complement of $A(X)$.

RCGs operate over *ranges* of the sentence, ie. substrings of words of the language. A *range* is formally defined, given a string $w = a_1 \dots a_n$, as a couple (i, j) , $0 \leq i \leq j \leq n$, and represents the substring $a_{i+1} \dots a_j$ in w . (i, j) is of size $j - i$ and can be empty (ϵ) if $i = j$. Consecutive ranges can then be concatenated: $(i, j) \bullet (j, k) = (i, k)$.

The arguments of a RCG clause may be bound to ranges of a given string w : $A(aX, bY) \rightarrow B(X, Y)$ can be instanciated to $A((i, j), (k, l)) \rightarrow B((i + 1, j), (k + 1, l))$ if the string w is such as $a_{i+1} = a$ and $a_{k+1} = b$.

Finally, a string w is produced by a given RGC G if the empty range ϵ can be derived from any valid instanciation of the start predicates, using the clauses in G , the *language* of G being the set of all of such strings.

1.2 Examples

The following RCG, from [Bou99a], defines the language $\{w^3, w \in \{a, b\}^*\}$:

$$\begin{aligned} S(XYZ) &\rightarrow A(X, Y, Z) \\ A(aX, aY, aZ) &\rightarrow A(X, Y, Z) \\ A(bX, bY, bZ) &\rightarrow A(X, Y, Z) \\ A(\varepsilon, \varepsilon, \varepsilon) &\rightarrow \varepsilon \end{aligned}$$

The sequence of derivations of *abaabaaba* being:

$$S(abaabaaba) \rightarrow A(aba, aba, aba) \rightarrow A(ba, ba, ba) \rightarrow A(a, a, a) \rightarrow A(\varepsilon, \varepsilon, \varepsilon) \rightarrow \varepsilon$$

One can also define the language $\{a^n b^n c^n d^n, n > 0\}$, using the following rules:

$$\begin{aligned} S(XYZT) &\rightarrow A(X, Y, Z, T) \\ A(aX, bY, cZ, dT) &\rightarrow B(X, Y, Z, T) \\ B(X, Y, Z, T) &\rightarrow A(X, Y, Z, T) \\ B(\varepsilon, \varepsilon, \varepsilon, \varepsilon) &\rightarrow \varepsilon \end{aligned}$$

Thus, *aabbccdd* would be derived as:

$$S(aabbccdd) \rightarrow A(aa, bb, cc, dd) \rightarrow B(a, b, c, d) \rightarrow A(a, b, c, d) \rightarrow B(\varepsilon, \varepsilon, \varepsilon, \varepsilon) \rightarrow \varepsilon$$

Note that these languages are not context-free, and often used to illustrate the limitations of formalisms akin to CFGs.

1.3 Completeness of the formalism

In [Bou99b], Pierre Boullier proves that the subclass of RCGs with predicates of at most one argument (1-RCGs) is, in itself, a formalism powerful enough to encompass all languages constructed by the intersection and complementation of context-free languages, and that can be used to solve some of the problems posed by NLP, while remaining as efficient as CFGs, ie. parsable in $O(n^3)$ time.

In [Bou98], however, the author goes on to prove that **RCGs can generate any language that is recognizable in deterministic polynomial time** (*PTIME*, which most probably includes the set of human-understandable languages), while keeping a reasonable parse time (polynomial with respect to the size of the text and linear to the size of the grammar). As such, he believes that RCGs can be easily used to accurately describe and compute human languages.

To illustrate his point, the author has written simple RCGs describing some non-trivial phenomena, thought to be difficult to compute, in [Bou99a]. Thus, he argues that large numbers in Mandarin Chinese and German scrambling are covered by his formalism.

1.4 Relationship to other formalisms

Many other formalisms have been introduced before RCGs, representing different language sets or solving different parsing problems. The most important work of Pierre Boullier has been to prove, for a number of those formalisms, that for every instance of such formalisms, there exists a weakly

equivalent¹ RCG; this equivalent RCG can, in each case, be constructed automatically, and has a parsing complexity equal or less than the one of its counterpart.

In [Bou98], Pierre Boullier describes the construction of weakly equivalent RCGs for CFGs, Linear Indexed Grammars, Tree Adjoining Grammars, Head Grammars, Coupled Context-Free Grammars, and Linear Context-Free Rewriting Systems. He also points out that, because of these constructions, RCGs can be thought of as a high-level implementation structure for any of those formalisms.

2 Additional material

In this section, we will discuss some scientific publications referring to RCGs, covering research done after their introduction.

2.1 Extensions and applications

On the theoretical side, in [Lju05], Peter Ljunglöf extends Parallel Multiple Context-Free Grammars with an *intersection* operation, and proves that the resulting formalism is weakly equivalent to RCGs. The work on RCGs is used to prove that these PMCFGs cover *PTIME*.

On the practical side, after formally defining RCGs, Pierre Boullier and other members of his team set out to apply them to common linguistic constructions.

In [SEG04], Benoît Sagot and Adil El Ghali describe how to interface an RCG and a knowledge base (implemented in Description Logics) in order to parse the sentence syntactically and semantically, constructing a logical representation at the same time. This dual approach could be a lot more efficient than other traditional logical representation formalisms.

In [Sag05], Benoît Sagot introduces a complete parsing system using only RCGs and allowing for analysis of syntactic, semantic and possibly other properties of the sentence; the reasoning being that an RCG can be constructed for the syntactic structure, and semantic properties (or other linguistic facts) can be considered as additional predicates, forming a Meta-RCG (that is itself equivalent, and thus parsable as, another RCG). Benoît Sagot illustrates this construction by the parsing of example sentences, using a work-in-progress grammar for French.

2.2 Possible limitations

In [Chi04], David Chiang suspects that formalisms built on the intersection of string languages (such as, he claims, RCGs), are not powerful enough to express the necessary facts on the structure they describe. This seems a bit far-fetched, considering that all of the properties of the CFGs are thoroughly formally proved in [Bou98], but Chiang does argue that their are constructions

¹Formalisms for syntactic analysis are *equivalent* when the language they generate are equal. They are said to be *strongly* equivalent if the derivation process used for a given sentence in one formalism can be inferred from the one used in the other, *weakly* so otherwise.

in German scrambling not covered by Boullier's example (in [Bou99a]). However, as we have not heard of a response from Pierre Boullier, whether these inconsistencies are inherent with the formalism, as Chiang claims, or simply with lack of care in the construction of the example grammar, remains to be seen.

This point shows, at the least, that the construction of an RCG is not trivial.

To sum up...

So far, RCGs seem to be good candidates for a complete NLP backbone. The formal results of Pierre Boullier's research show that instances of the formalism would probably be sufficient, and efficient enough, to parse any linguistic property over any known, human-understandable language... once they are constructed.

Also, despite the author's claim that a RCG can express the set of all possible derivation trees for a sentence, he does not seem to give a method for extracting the actual tree used for the syntactic analysis of a given sentence, which would could be used for further grammatical and semantic parsing.

On the other hand, examples having been both devised and implemented to use the formalism, we know they are useable in practical computations. Of course, those practical implementations remain experimental, especially for semantic analysis, and are still to be investigated.

Yet, on the whole, it appears that Boullier's proposal is formally sound.

References

- [BBDV01] François Barthélemy, Pierre Boullier, Philippe Deschamp, and Éric Villemonte de la Clergerie. Guided parsing of range concatenation languages. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL'01)*, pages 42–49, July 2001.
- [Bou98] Pierre Boullier. Proposal for a Natural Language Processing Syntactic Backbone. Technical report, INRIA, January 1998.
- [Bou99a] Pierre Boullier. Chinese numbers, mix, scrambling, and range concatenation grammars. In *Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics (EACL'99)*, pages 53–60, Bergen, Norway, June 1999.
- [Bou99b] Pierre Boullier. A cubic time extension of context-free grammars. In *Sixth Meeting on Mathematics of Language (MOL6)*, pages 37–50, University of Central Florida, Orlando, Florida, USA, July 1999.
- [Bou01] Pierre Boullier. From contextual grammars to range concatenation grammars. In *Sixth Conference on Formal Grammar and Seventh Conference of Mathematics of Language (FG/MOL'01)*, University of Helsinki, Helsinki, Finland, August 2001.
- [Chi04] David Chiang. Uses and abuses of intersected languages. In *Seventh International Workshop on Tree Adjoining Grammar and Related Formalisms*, May 2004.
- [Lju05] Peter Ljunglöf. A Polynomial Time Extension of Parallel Multiple Context-Free Grammar. In *LACL 2005*, 2005.
- [Sag05] Benoît Sagot. Linguistic facts as predicates over ranges of the sentence. In *Proceedings of LACL'05*, pages 271–286, Bordeaux, France, April 2005.
- [SB04] Benoît Sagot and Pierre Boullier. Les RCG comme formalisme grammatical pour la linguistique. In *Actes de TALN'04*, pages 403–412, Fès, Maroc, 2004.
- [SEG04] Benoît Sagot and Adil El Ghali. Coupling grammar and knowledge base: Range concatenation grammars and description logics. In *Proceedings of TSD'04*, Brno, Tchéquie, 2004.