Deductive Parsing with Interaction Grammars

Joseph Le Roux
NCLT, School of Computing,
Dublin City University
jleroux@computing.dcu.ie

Abstract

We present a parsing algorithm for Interaction Grammars using the deductive parsing framework. This approach brings new perspectives to this problem, departing from previous methods which rely on constraint-solving techniques.

1 Introduction

An Interaction Grammar (IG) (Guillaume and Perrier, 2008) is a lexicalized grammatical formalism that primarily focuses on valency, explicitly expressed using polarities decorating syntagms. These polarities and the use of underspecified structures naturally lead parsing to be viewed as a constraint-solving problem – for example (Bonfante et al.) reduce the parsing problem to a graph-rewriting problem in (2003) .

However, in this article we depart from this approach and present an algorithm close to (Earley, 1970) for context-free grammars. We introduce this algorithm using the standard framework of deductive parsing (Shieber et al., 1995).

This article is organised as follows: we first present IGs (section 2), then we describe the algorithm (section 3). Finally we discuss some technical points and conclude (sections 4 and 5).

2 Interaction Grammars

We briefly introduce IGs as in (Guillaume and Perrier, 2008)[1]. However, we omit polarized feature structures, for the sake of exposition.

2.1 Polarized Tree Descriptions

The structures associated with words by the lexicon are Polarized Tree Descriptions (PTDs). They represent fragments of parse trees. The nodes of these structures are labelled with a category and a polarity. IGs use 4 polarities, $\mathbb{P} = \{\rightarrow, \leftarrow, =, \sim\}$, namely positive, negative, neutral and virtual.

A multiset of polarities is *superposable*[2] if it contains at most one $\rightarrow$ and at most one $\leftarrow$.

A multiset of polarities is *saturated* if it contains either (1) one $\rightarrow$, one $\leftarrow$ and any number of $\sim$ and $=$, or (2) zero $\rightarrow$, zero $\leftarrow$, any number of $\sim$ and at least one $=$.

The two previous definitions can be extended to nodes: a multiset of nodes is saturated (resp. superposable) if all the elements have the same category and if the induced multiset of polarities is saturated (resp. superposable).

A PTD is a DAG with four binary relations: the immediate dominance $>$, the general dominance $>^*$, the immediate precedence $\prec$ and the general precedence $\prec^+$. A valid PTD is a PTD where (1) $>$ and $>^*$ define a tree structure[3] , (2) $\prec$ and $\prec^+$ are restricted to couples of nodes having the same ancestor by $>$, and (3) one leaf is the anchor. In the rest of this paper, all PTDs will be valid.

We now introduce some notations : if $n >^* m$, we say that m is constrained by n and for a set of nodes $\mathcal{N}$, we define $\mathcal{N}^{\Cross} = \{N | \exists M \in \mathcal{N}, M \Cross N\}$ where $\Cross$ is a binary relation.

2.2 Grammars

An IG is a tuple $\mathcal{G} = \{\Sigma, \mathbb{C}, S, \mathcal{P}, phon\}$, where Σ is the terminal alphabet, $\mathbb{C}$ the non-terminal alphabet, $S \in \mathbb{C}$ the initial symbol, $\mathcal{P}$ is a set of PTDs with node labels in $\mathbb{C} \times \mathbb{P}$, and $phon$ is a function from anchors in $\mathcal{P}$ to Σ.

The structure obtained from parsing is a *syntactic tree*, a totally ordered tree in which all nodes are labelled with a non-terminal. We call $lab(A)$ the label of node A. If a leaf L is labelled with a terminal, this terminal is denoted $word(L)$.

[1] This paper also discusses the linguistic motivations behind IGs.

[2] This name comes from the *superposition* introduced in previous presentations of IGs.

[3] For readers familiar with D-Tree Grammars (Rambow et al., 1995), $>$ adds an i-edge while $>^*$ adds a d-edge.

Proceedings of the 11th International Conference on Parsing Technologies (IWPT), pages 65–68,
Paris, October 2009.

We will write $M \gg N$ if the node M is the mother of N and $N \gg [N_1, \ldots, N_k]$ if the N is the mother of the ordered list of nodes $[N_1, \ldots, N_k]$. The order between siblings can also be expressed using the relation $\prec\!\!\prec$: $M \prec\!\!\prec N$ means that N is the immediate successor of M. $\prec\!\!\prec^+$ is the transitive closure of $\prec\!\!\prec$ and $\gg^*$ the reflexive transitive closure of $\gg$.

We define the phonological projection PP of a node as : $PP(M) = [t]$ if $M \gg []$ and $word(M) = t$, or $PP(M) = [PP(N_1) \ldots PP(N_k)]$ if $M \gg [N_1, \ldots, N_k]$

A syntactic tree T is a *model* for a multiset $\mathcal{D}$ of PTDs if there exists a total function I from nodes in $\mathcal{D}$ ($\mathcal{ND}$) to nodes in T ($\mathcal{NT}$). I must respect the following conditions, where variables M, N range over $\mathcal{ND}$ and A, B over $\mathcal{NT}$:

1. $I^{-1}(A)$ is saturated and non-empty.
2. if $M > N$ then $I(M) \gg I(N)$
3. if $M >^* N$ then $I(M) \gg^* I(N)$
4. if $M \prec N$ then $I(M) \prec\!\!\prec I(N)$
5. if $M \prec^+ N$ then $I(M) \prec\!\!\prec^+ I(N)$
6. if $A \gg B$ then there exists $M \in I^{-1}(A)$ and $N \in I^{-1}(B)$ such that $M > N$
7. $lab(A) = lab(M)$ for all $M \in I^{-1}(A)$
8. if $phon(M) = w$ then $PP(I(M)) = [w]$

Given an IG $\mathcal{G} = \{\Sigma, \mathbb{C}, S, \mathcal{P}, phon\}$ and a sentence $s = w_1, \ldots, w_n$ in Σ^*, a syntactic tree T is a parse tree for s if there exists a multiset of PTDs $\mathcal{D}$ from $\mathcal{P}$ such that the root node R of T is labelled with S and $PP(R) = [w_1, \ldots, w_n]$. The language generated by $\mathcal{G}$ is the set of strings in Σ^* for which there is a parse tree.

3 Parsing Algorithm

We use the deductive parsing framework (Shieber et al., 1995). A state of the parser is encoded as an item, created by applying deductive rules. Our algorithm resembles the Earley algorithm for CFGs and uses the same rules : prediction, scanning and completion.

3.1 Items

Items $[A(H, N, F) \rightarrow \alpha \bullet \beta, i, j, (O, U, D)]$ consist of a dotted rule, 2 position indexes and a 3-tuple of sets of constrained nodes.

The dotted rule $A(H, N, F) \rightarrow \alpha \bullet \beta$ means that there exists a node A in the parse tree with antecedents $H \cup N \cup F$. Elements of the sequence α are also nodes of the parse tree. For the sequence β, the elements have the form $B_k(H_k)$ where B_k is a node of the parse tree and H_k is a subset of its antecedents, the predicted antecedents.

This item asserts that a syntactic tree can be partially built from the input grammar and sentence, that contains $A \gg [A_1 \ldots A_k B_1 \ldots B_l]$ and that $PP(A_1) \circ \cdots \circ PP(A_k) = [m_{i+1} \ldots m_j]$.

The proper use of constrained nodes is managed by O, U and D:

- Nodes in D are available in prediction to find antecedents for new parse tree nodes.
- Nodes in O must be used in a sub-parse. To use an item as a completer, O must be empty.
- U contains constrained nodes that have been used in a prediction, and for which the constraining nodes have not been completed yet.

Moreover, we will use 3 additional symbols: $\top$ as the left-hand side of the axiom item which can be seen as a dummy root, and $\blacksquare$ or $\blacklozenge$ that mark items for which prediction is not terminated.

We will need sequences of antecedents that respect the order relations of an IG. Given a set of nodes $\mathcal{N}$, we define the set of all these orderings:

$ord(\mathcal{N}) = \{[\mathcal{N}_1 \ldots \mathcal{N}_k] |$
$(\mathcal{N}_i)_{1 \leq i \leq k}$ is a partition of $\mathcal{N} \wedge$
$1 \leq i \leq k, \mathcal{N}_i$ is superposable $\wedge$
if $n_1, n_2 \in \mathcal{N}$ and $n_1 \prec n_2$ then
$\quad \exists 1 \leq j < k$ s.t. $n_1 \in \mathcal{N}_j$ and $n_2 \in \mathcal{N}_{j+1} \wedge$
if $n_1, n_2 \in \mathcal{N}$ and $n_1 \prec^+ n_2$ then
$\quad \exists 1 \leq i < j \leq k$ s.t. $n_1 \in \mathcal{N}_i$ and $n_2 \in \mathcal{N}_j\}$

3.2 Deductive Rules

In this section, we assume an input sentence $s = w_1, \ldots, w_n$ and a IG $\mathcal{G} = \{\Sigma, \mathbb{C}, S, \mathcal{P}, phon\}$.

Axiom This rule creates the first item. It prepares the prediction of a node of category S starting at position 0 without constrained nodes.

$$\frac{}{[\top \rightarrow \bullet S(\emptyset), 0, 0, (\emptyset, \emptyset, \emptyset)]}\, ax$$

Prediction This rule initializes a sub-parse. We divide it in three in order to introduce the different constraints one at a time.

$$\frac{[A(H,N,F) \rightarrow \alpha \bullet C(H_C)\beta, i, j, (O,U,D)]}{[C(H_C,\emptyset,\emptyset) \rightarrow \blacksquare, j, j, (\emptyset,U,D \cup O)]} p_1$$

In this first step, we initialize a new sub-parse at the current position j where C will be the predicted node that we want to find antecedents for. If some antecedents H_C have already been predicted we use them. The nodes in O, that must be used in one of the sub-parse of A, become available as possible antecedents for C.

$$\frac{[C(H_C,\emptyset,\emptyset) \rightarrow \blacksquare, j, j, (\emptyset,U_1,D_1)]}{[C(H_C,N_C,\emptyset) \rightarrow \blacklozenge, j, j, (\emptyset,U_2,D_2)]} p_2$$

$$\begin{cases} H_C \cup N_C \neq \emptyset \\ H_C \cup N_C \text{ is superposable} \\ N_C \subset D_1 \cup \text{ roots}(\mathcal{P}) \\ D_2 = D_1 - N_C \\ U_2 = U_1 \cup (D_1 \cap N_C) \end{cases}$$

In this second step, new antecedents for C are added from the set N_C, chosen among available nodes in D_1 and root nodes from the PTDs of the grammar. The 3 node sets are then updated. Constrained nodes that have been chosen as antecedents for C are not available anymore and are added to the set of used constrained nodes.

$$\frac{[C(H_C,N_C,\emptyset) \rightarrow \blacklozenge, j, j, (\emptyset,U,D)]}{[C(H_C,N_C,F_C) \rightarrow \bullet\gamma, j, j, (O,U,D)]} p_3$$

$$\begin{cases} H_C \cup N_C \cup F_C \text{ is saturated} \\ \gamma \in ord((H_C \cup N_C \cup F_C)^{>}) \\ F_C = \bigcup_i Q_i, Q_0 \subseteq (H_C \cup N_C)^{>^*}, Q_{i+1} \subseteq Q_i^{>^*} \\ O = (H_C \cup N_C \cup F_C)^{>^*} - F_C \\ \text{no anchor node in } H_C \cup N_C \cup F_C \end{cases}$$

In this last step of prediction, we can choose new antecedents for C among nodes constrained by antecedents already chosen in the previous steps in order to saturate them. This choice is recursive : each added antecedent triggers the possibility of choosing the nodes it constrains. The second part of this step consists of predicting the shape of the tree. We need to order and superpose the daughter nodes of the antecedents in such a way that ordering relations in PTDs are respected: an element of $ord((H_C \cup N_C \cup F_C)^{>})$ is chosen.

Finally, the nodes that must be used in a sub-parse are the ones that are constrained by antecedents of C and not antecedents themselves.

Scan This is the rule that checks predictions against the input string. It is similar to the previous rule, but one (and only one) of the antecedents must be an anchor.

$$\frac{[C(H_C,N_C,\emptyset) \rightarrow \blacklozenge, j, j, (\emptyset,U,D)]}{[C(H_C,N_C,F_C) \rightarrow \bullet, j, j+1, (\emptyset,U,D)]} s$$

$$\begin{cases} H_C \cup N_C \cup F_C \text{ is saturated} \\ (H_C \cup N_C \cup F_C)^{>} = \emptyset \\ F_C = \bigcup_i Q_i, Q_0 \subseteq (H_C \cup N_C)^{>^*}, Q_{i+1} \subseteq Q_i^{>^*} \\ (H_C \cup N_C \cup F_C)^{>^*} - F_C = \emptyset \\ \text{one anchor } a \text{ in } H_C \cup N_C \cup F_C \\ phon(a) = w_{j+1} \end{cases}$$

If the expected terminal is read on the input string, parsing can proceed. Note that antecedents for C should not constrain nodes that are not antecedents of C themselves.

Completion This rule extends a parse by combining it with a complete sub-parse.

$$\frac{\begin{array}{l}[A(H,N,F) \rightarrow \alpha \bullet C(H_c)\beta, i, j, (O_1,U_1,D_1)] \\ [C(H_C,N_C,F_C) \rightarrow \gamma\bullet, j, k, (\emptyset,U_2,D_2)]\end{array}}{[A(H,N,F) \rightarrow \alpha C \bullet \beta, i, k, (O_3,U_3,D_3)]} c$$

$$\begin{cases} N_C \subseteq D_1 \cup O_1 \cup \mathcal{P} \\ D_2 \subseteq (D_1 \cup O_1) - N_C \\ U_1 \subseteq U_2 \\ O_3 = O_1 - U_2 \\ D_3 = D_1 - U_2 \\ U_3 = U_2 - O_1 \end{cases}$$

We have to make sure that the second hypothesis is a sub-parse for the first : (1) the set of available nodes in the sub-parse must be a subset of the available nodes for current parse, (2) the set of used nodes in the main parse must be a subset of the used nodes in the sub-parse and (3) used nodes constrained by the first hypothesis disappear.

Goal Parsing is successful if the following item is created : $[\top \rightarrow S\bullet, 0, n, (\emptyset,\emptyset,\emptyset)]$.

4 Discussion

4.1 Consistency and completeness

An item $[A(H,N,F) \rightarrow \alpha \bullet \beta, i, j, (O,U,D)]$ asserts the following invariants :

- A and the elements α_l of α are models for saturated sets of nodes. Conditions 1, 7 and 3 (reflexive case) of a model are respected.
- Elements β_k of β are superposable. Then we have $\beta_k \subseteq (A^{-1})^{>}$ (conditions 2 and 6).
- the sequence $\alpha\beta$ is compatible with the order relations from the PTDs (conditions 4 and 5).
- $PP(\alpha_1) \circ \ldots \circ PP(\alpha_l) = [w_{i+1} \cdots w_j]$
- a node N in U is a constrained node in relation $>^*$ with a node such that condition 3 holds.

These invariants can be checked by induction on rules. Hence, such an item asserts there exists a function J from the nodes of a subset of the PTDs of an IG to a syntactic tree with its root labelled by S and phonological projection $w_1 \ldots w_j$. This function has the same properties as the function I for models but conditions 2 to 5 only apply if both nodes are in the domain of J. The parsing process extends the domain until (1) all the nodes of each PTD selected are used and (2) the input string has been read completely. Then J defines a syntactic tree which is a parse tree.

4.2 Sources of non-determinism

The parsing problem in IGs is a NP-hard problem (Bonfante et al., 2003). Our presentation lets us see several sources of non-determinism.

In p_2, new antecedents are chosen among available nodes and root nodes of PTDs from the input grammar. There is an exponential number of such choices. However, IGs are lexicalized : only PTDs associated with a word in the sentence will be used and efficient lexical filters have been developed for IGs (Bonfante et al., 2006) that drastically decrease the number of PTDs to consider.

In p_3 and s, constrained nodes can be chosen as antecedents (nodes in F_C). There is again an exponential number of such choices. But in existing IGs, nodes have at most one successor by $>^*$ and there is no chain of nodes in relation by $>^*$. Consequently, $|F_C|$ can be bounded by $|H_C \cup N_C|$.

In p_3, daughters must be partitioned. Instead of building all these partitions in p_3 and generating many useless items, one can think of a lazy approach like the one proposed by (Nederhof et al., 2003) for pomset-CFGS.

It can be noticed that the completion rule, while having the most positional indexes, is not a particular source of non-determinism.

5 Conclusion

We presented a parsing algorithm for IGs. Although we used a simplified version without polarized feature structures, adding a unification mechanism shouldn't be an issue. The novelty of this presentation is the use of deductive parsing for a formalism developed in the model-theoretic framework (Pullum and Scholz, 2001).

This change of perspective provides new insights on the causes of non-determinism. It is a first step to a precise complexity study of the problem. In the future, it will be interesting to search for algorithmical approximations to improve efficiency. Another way to overcome NP-hardness is to restrict superpositions, as in (k-)TT-MCTAGs (Kallmeyer and Parmentier, 2008).

References

G. Bonfante, B. Guillaume, and G. Perrier. 2003. Analyse syntaxique électrostatique. *Traitement Automatique des Langues*, 44(3).

G. Bonfante, J. Le Roux, and G. Perrier. 2006. Lexical disambiguation with polarities and automata. In *Proceedings of CIAA*.

J. Earley. 1970. An efficient context-free parsing algorithm. *Communications of the ACM*, 13(2):94–102.

B. Guillaume and G. Perrier. 2008. Interaction Grammars. Research Report RR-6621, INRIA.

L. Kallmeyer and Y. Parmentier. 2008. On the relation between TT-MCTAG and RCG. In *Proceedings of LATA*.

M.J. Nederhof, G. Satta, and S. Shieber. 2003. Partially ordered multiset context-free grammars and ID/LP parsing. In *Proceedings of IWPT*.

G. Pullum and B. Scholz. 2001. On the distinction between model-theoretic and generative-enumerative syntactic frameworks. In *Proccedings of LACL*.

O. Rambow, K. Vijay-Shanker, and D. Weir. 1995. D-tree grammars. In *Proceedings of ACL*.

S. Shieber, Y. Schabes, and F. Pereira. 1995. Principles and implementation of deductive parsing. *Journal of Logic Programming*, 24(1–2):3–36.