

## A French Interaction Grammar

Guy Perrier

► **To cite this version:**

Guy Perrier. A French Interaction Grammar. International Conference on Recent Advances in Natural Language Processing - RANLP 2007, IPP & BAS & ACL-Bulgaria, Sep 2007, Borovets, Bulgaria. pp.463-467. inria-00184108

**HAL Id: inria-00184108**

**<https://hal.inria.fr/inria-00184108>**

Submitted on 30 Oct 2007

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A French Interaction Grammar

Guy Perrier  
LORIA - universit  Nancy 2  
BP 239  
54506 Vandœuvre-lès-Nancy cedex - France  
*perrier@loria.fr*

## Abstract

We present a relatively large coverage French grammar written with the formalism of Interaction Grammars. This formalism combines two key ideas: the grammar is viewed as a constraint system, which is expressed through the notion of tree description, and the resource sensitivity of natural languages is used as a syntactic composition principle by means of a system of polarities. We give an outline of the expressivity of the formalism by modelling significant linguistic phenomena and we show that the grammar architecture provides for re-usability and tractability, which is crucial for building large coverage resources: a modular source grammar is distinguished from the object grammar which results from the compilation of the first one, and the lexicon is independent of the grammar. Finally, we present the results of an evaluation of the grammar achieved with the LEOPAR parser with a test suite of sentences.

Although we use an original formalism, we are concerned with re-usability, which is expressed in two ways. Like with for programming languages, we distinguish two levels in the grammar. The *source grammar* aims at representing linguistic generalisations and it is written by a human, while the *object grammar* is directly usable by a NLP system and results from the compilation of the first one. In our case, we used XMG [2], a tool devoted to this goal. XMG provides a high level language for writing a source grammar and a compiler which translates this grammar into an operational object grammar. The grammar is also designed in such a way that it can be linked with a lexicon independent of the formalism, where entries appear as feature structures.

The goal of the article is to show that it is possible to build realistic grammatical resources, which integrate a refined linguistic knowledge with a large coverage, and for this, we have chosen an experimental approach with the construction of a French grammar.

## Keywords

Syntax, grammatical formalism, tree description, polarity, categorial grammar, unification grammar, interaction grammar

## 1 Introduction

The goal of our work is to model natural languages starting from linguistic knowledge and giving a central role to experimentation. For this, we need to express the linguistic knowledge by means of grammars and lexicons with the largest possible coverage: grammars have to represent all common linguistic phenomena and lexicons have to include the most frequent words with their most frequent use. As everyone knows, building such resources is a very hard task.

Firstly, we have to choose the formalism to represent the grammar. Currently, there is no leader among the formalisms used in the scientific community. Each of the most popular formalisms has its own advantages and drawbacks. We have designed a new formalism, Interaction Grammars (IG), the goal of which is to synthesize two key ideas, expressed in two kinds of formalisms up to now: using the resource sensitivity of natural languages as a principle of syntactic composition, which is a characteristic feature of Categorical Grammars (CG) [9], and viewing grammars as constraint systems, which is a feature of unification grammars such as LFG [1] or HPSG [11].

## 2 Interaction Grammars

IG [5, 6] is a grammatical formalism which is devoted to the syntax and semantics of natural languages and which uses two notions: *tree description* and *polarity*.

### 2.1 Tree Descriptions

In a derivational view of the syntax of natural languages, the basic objects are trees and they are composed together in a more or less sophisticated way: by substitution in Context Free Grammars, by adjunction in Tree Adjoining Grammars, by application and abstraction in Categorical Grammars ... Taking our view from the Model Theory [7], we do not directly manipulate trees but properties which are used to describe them, in other words tree descriptions [10]. This approach is very flexible as it allows the expression of elementary properties in a totally independent way, as they can be freely combined.

A tree description can be viewed either as an underspecified tree, or as the specification of a tree family, each tree being a model of the specification. Figure 1 gives an example of a tree description, which is associated with the relative pronoun *qui* (who), used inside a prepositional complement. This use gives rise to the phenomenon of  *pied piping* as the following example illustrates: *Jean [à la femme de **qui**] Pierre sait qu'on a présenté Marie □, est ingénieur (Jean [to whose wife]*

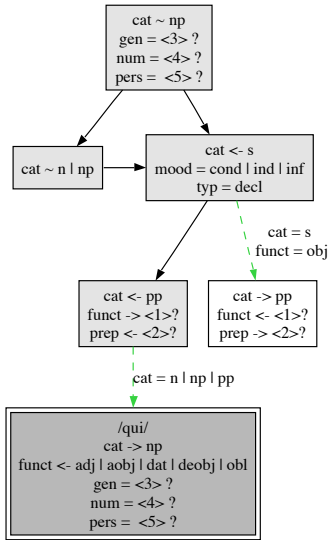


Fig. 1: Tree description associated with the relative pronoun *qui* used inside a prepositional complement

*Pierre knows someone presented Marie*  $\square$ , *is an engineer*). This example is covered by the description of figure 1.<sup>1</sup>

A tree description is a finite set of nodes structured by two kinds of relations: *dominance* and *precedence*. Dominance relations can be immediate or large (respectively solid and dashed down arrows in figure 1). Constraints can be put on intermediate nodes for large dominance relations. Precedence relations (horizontal arrows in figure 1) can also be immediate or large.

Nodes, which represent constituents, are labelled with features describing their morpho-syntactic properties. Feature values are atoms or atom disjunctions and they can be shared with the help of a co-indexation mechanism.<sup>2</sup> Nodes can be *Empty* (the white box in figure 1) or *Full*, according to whether they have an empty phonological form or not. Full nodes can be *Anchors* (the dark box in figure 1), if they anchor a word of the language.

## 2.2 Polarities

Polarities are used to express the saturation state of syntactic trees. They are attached to features that label description nodes with the following meaning:

- a positive feature  $t \rightarrow v$  expresses an available resource, which must be consumed;
- a negative feature  $t \leftarrow v$  expresses an expected resource, which must be provided; it is the dual of a positive feature;
- a neutral feature  $t = v$  expresses a linguistic property that is not a consumable resource.

<sup>1</sup> The extracted prepositional phrase is put between square brackets and its trace in the relative clause is represented by the  $\square$  symbol.

<sup>2</sup> When two features share the same value, a common index  $\langle n \rangle$  is put before their values. When a feature value is the disjunction of all elements of a domain, this value is denoted with "?".

- a virtual feature  $t \sim v$  expresses a linguistic property that needs to be realised by combining with an actual feature (an actual feature is a positive, negative or neutral feature).

In figure 1, the empty node representing the trace of the prepositional phrase extracted from the relative clause carries a positive feature  $cat \rightarrow pp$  and a negative feature  $funct \leftarrow \langle 1 \rangle?$ , which means that this node provides a prepositional phrase that needs to receive a syntactic function. The tree root carries a virtual feature  $cat \sim np$  which means that the node represents a virtual noun phrase which has to combine with an actual noun phrase.

The descriptions labelled with polarised feature structures are called *polarised tree descriptions (PTDs)* in the rest of the article.

## 2.3 Grammars as constraint systems

A particular interaction grammar is defined by a finite set of elementary PTDs, which generates a tree language. A tree belongs to the language if it is a model of a finite set of elementary PTDs with two properties:

- It is *saturated*: every positive feature  $t \rightarrow v$  is matched with its dual feature  $t \leftarrow v$  in the model and vice versa. Moreover, every virtual feature has to find an actual corresponding feature in the model.
- It is *minimal*: the model has to add a minimum of information to the initial descriptions (it cannot add immediate dominance relations or features that do not exist in the initial descriptions).

Then, parsing reduces to the resolution of a constraint system. It consists of building all saturated and minimal models of a finite set of elementary PTDs. In practice, our grammar is totally lexicalized: each elementary PTD has a unique anchor, which is used for linking the description with a word of the language. In this way, in the parsing of a sentence, it is possible to select the only PTDs that are anchored by words of the sentence. The set of PTDs being selected, the building of a saturated and minimal model is performed step by step by means of a merging operation between nodes, which is guided by one of the following constraints:

- neutralise a positive feature with a negative feature having the same name and carrying a value unifiable with the value of the first feature;
- realise a virtual feature by combining it with an actual feature (a positive, negative or neutral feature) having the same name and carrying a value unifying with the value of the first feature.

The constraints of the description interact with node merging to entail a partial superposition of their contexts represented by the tree fragments in which they are situated. To summarise, IG combine the strong points of two families of formalisms: the flexibility of *Unification Grammars* and the saturation control of *Categorial Grammars*.

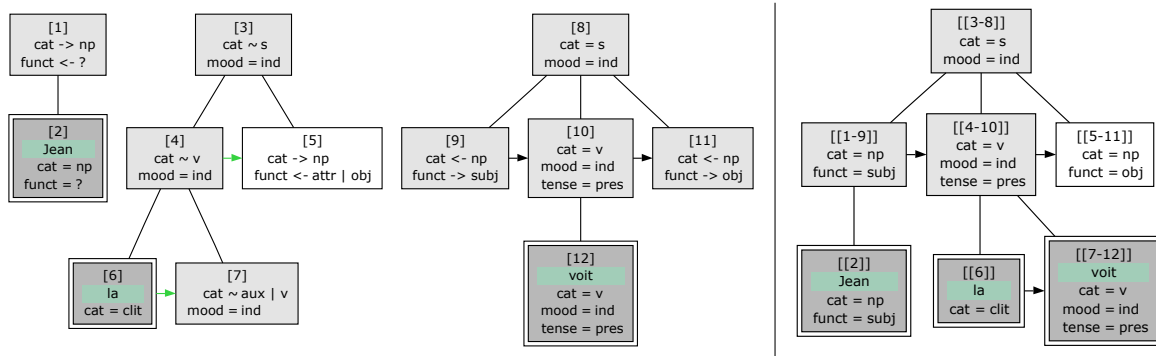


Fig. 2: PTD associated with the sentence *Jean la voit* and its minimal saturated model

Figure 2 presents an example of parsing for the sentence *Jean la voit* (*Jean sees her*).<sup>3</sup> The left side shows the set of initial PTDs associated with the sentence by the grammar. The grammar being lexicalized, each PTD is anchored by a word of the sentence and it has been extracted from a lexicon. These PTDs have been gathered in a unique PTD and precedence relations between anchors have been added to express word order in the sentence. These relations do not appear in figure 2.

The computation of the model shown on the right side of figure 2 from the initial description shown on the left side is performed by a sequence of 3 node mergings.<sup>4</sup> The interaction of tree constraints with these mergings entails two other mergings and a partial tree superposition.

### 3 The expressivity of Interaction Grammars

In the limits of this article, we have chosen to illustrate three aspects which are especially significant.

#### 3.1 Unbounded dependencies and underspecified dominance relations

Underspecified dominance relations are used to represent unbounded dependencies and the feature structures that can be associated with these relations allow the expression of constraints on these dependencies: barriers to extraction for instance.

Relative pronouns, such as *qui* or *lequel*, give rise to pied piping as the following sentence shows: *Jean [dans l'entreprise de **qui**] Marie sait que l'ingénieur travaille □, est malade* (*Jean [in whose firm] Marie knows that the engineer works □, is ill*):

- There is a first unbounded dependency between the verb *travaille* and its extracted complement *dans l'entreprise de qui*. The trace of the extracted complement is denoted by the □ symbol. The dependency is modelled in the PTD associated with the *qui* relative pronoun represented

in figure 1 by means of an underspecified dominance relation. The constraint linked to this dominance relation expresses that the dependency of the prepositional phrase on the verb of which it is the complement can only cross an unspecified sequence of embedded object clauses.

- Inside the prepositional phrase, there is a second unbounded dependency between the head of the constituent and the *qui* relative pronoun, which can be embedded arbitrarily deeply. This dependency is also represented in figure 1 with an underspecified dominance relation and the linked constraint expresses that all embedded constituents from the prepositional phrase to the *qui* relative pronoun are common nouns, noun phrases or prepositional phrases.

#### 3.2 Polarities used for modelling negation

In French, negation can be expressed with the help of the particle *ne* paired with a specific determiner, pronoun or adverb. The position of the particle *ne* is fixed before an inflected verb but the second component of the pair, if it is a determiner like *aucun* or a pronoun like *personne*, can have a relatively free position in the sentence, as illustrated by the following examples:

- Jean ne parle à aucun collègue* (*Jean speaks to no colleague*).
- Jean ne parle à la femme d'aucun collègue* (*Jean speaks to the wife of no colleague*).
- Aucun collègue de Jean ne parle à sa femme* (*No colleague of John's speaks to his wife*).

As figure 3 shows, the pairing of *ne* with *aucun* is expressed with a *neg* polarised feature attached to the node representing the maximal projection of the verbal kernel: *aucun* is waiting for such a feature, which will be provided by *ne*. The relatively free position of *aucun* is expressed by an underspecified dominance relation of the node representing the clause on the noun phrase that it introduces. The constraint linked to this dominance relation expresses the fact that *aucun* can only introduce arguments of the verbal head of the sentence or complements of these arguments.

<sup>3</sup> We have simplified the figure by ignoring agreement features.

<sup>4</sup> The head of each node includes the numbers of the nodes from the initial PTD which have been merged.

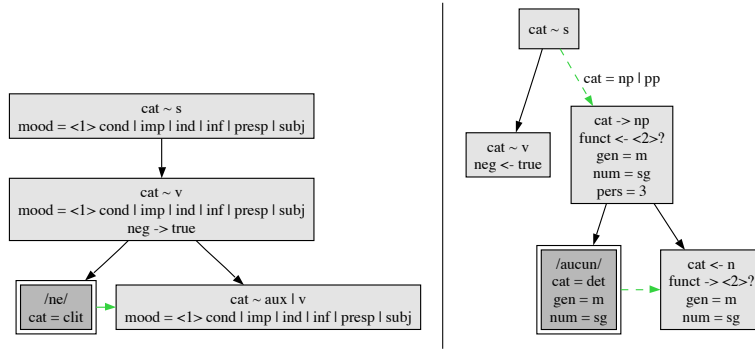


Fig. 3: PTDs respectively associated with the particle *ne* and the determiner *aucun*

### 3.3 The adjunction of modifiers by means of virtual polarities

In French, the position of adjuncts in the sentence is relatively free, as illustrated by the following example. In the sentence  $\square$  *Jean*  $\square$  *va*  $\square$  *rendre visite*  $\square$  *à Marie*  $\square$  (*Jean is going to visit Marie*), the sentence modifier *le soir* (*tonight*) can appear at any position marked with a  $\square$  symbol, according to different communicative goals.

The *virtual* polarity  $f \sim v$  did not exist in the previous version of IG [6]. Modifier adjunction was performed by addition of a new level in the syntactic tree of the constituent being modified. Sometimes, introducing an additional level is justified linguistically, but in most cases it introduces artificial complexity and ambiguity. Taking again an idea of [4], with his system of black and white polarities, we have introduced virtual polarities. This allows a modifier to be added as a new daughter of the node that it modifies without changing the rest of the syntactic tree, in which the modified node is situated. This operation is called *sister adjunction* and it is used in some formalisms: dependency grammars, description substitution grammars [8]. This way of modelling modifiers is more flexible and it allows the previous examples to be treated without difficulty, including parenthetical clauses.

## 4 The architecture of the grammar

### 4.1 The modular organisation of the grammar

The grammar has been built with the XMG tool [2], which allows grammars to be written with a high level of abstraction in a modular setting and to be compiled into low level grammars, usable by NLP systems.

A grammar is organised as a class hierarchy by means of two composition operations: *conjunction* and *disjunction*. It is also structured according to several dimensions, which are present in all classes. Our grammar uses only two dimensions: the first one is the syntactic dimension, where objects are PTDs, and the second one is the dimension of the interface with the lexicon, where objects are feature structures.

To define the conjunction of two classes one needs to

specify the way of combining the components of each dimension: for the syntactic dimension, PTD union is performed; for the lexicon interface dimension, it is realised as unification between feature structures.

The current grammar is composed of 448 classes, including 121 terminal classes, which are compiled into 2059 PTDs. These classes are ranked by family. Some classes from a family can be used in the definition of classes belonging to another family. This is the case for instance for the *Complement* family, which include classes related to complements of predicative structures. It is used by three other families: *Adjective*, *Noun* and *VerbDiathese*, which respectively refer to adjectives, nouns and various verbal diatheses.

### 4.2 The link with a lexicon independent of the formalism

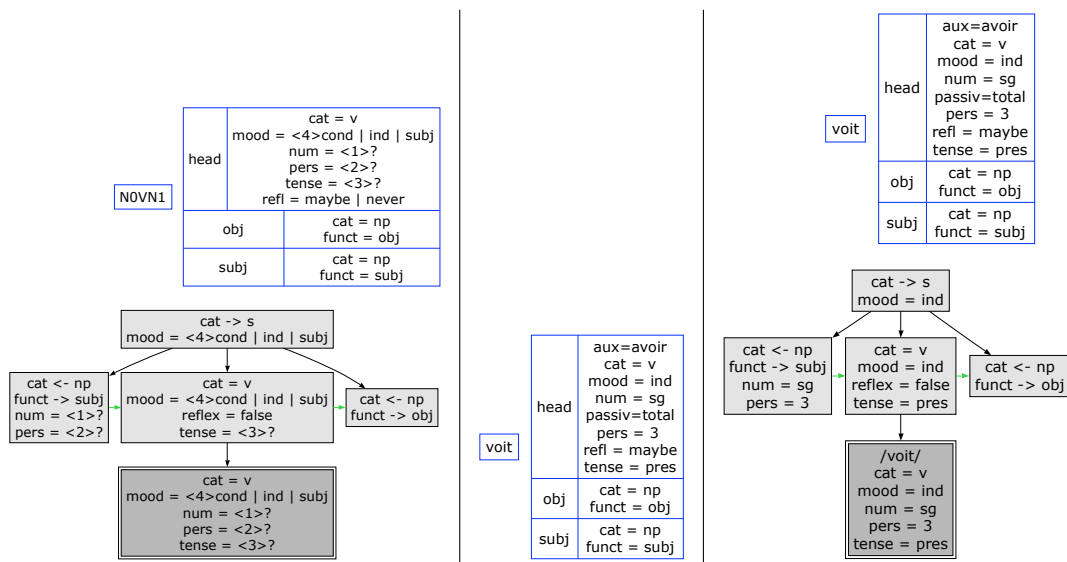
The grammar, in its current setting, is totally lexicalised: each elementary PTD of the grammar has a unique anchor node intended to be linked with a word of the language. Each PTD is associated to a feature structure, which describes a syntactic frame corresponding to words able to anchor the PTD, the description being independent of the formalism. This feature structure constitutes the PTD interface with the lexicon.

The set of features used in the interfaces differs from that used in PTDs because they do not play the same role: they do not aim at describing syntactic structures but they are used for describing the morpho-syntactic properties of the words of the language in a way independent of the formalism.

The left side of figure 4 shows a non anchored PTD describing the syntactic behaviour of a transitive verb in the active voice. The PTD is accompanied by its interface, which is a two level feature structure.

The lexicon associates words of the language to syntactic frames in a form identical to the PTD interfaces. For instance, the central part of figure 4 shows a lexical entry for the verb *voit* in its transitive use.

The PTD anchoring is then performed by unification of the PTD interfaces with the compatible entries of the lexicon. Figure 4 on its right side shows a PTD anchored by the transitive verb *voit*. This PTD comes from the unification between the lexical entry for *voit* presented in the center of the figure and the interface of the non anchored PTD on the left side of the figure.



**Fig. 4:** From left to right, a non anchored PTD describing the syntactic behaviour of a transitive verb in the active voice, a lexical entry for the transitive verb *voit* and the PTD after anchoring with the verb *voit*

## 5 Evaluation on a sentence test suite

Our goal is to evaluate the coverage of our grammar in the most detailed manner. The least costly way of doing this is to use the grammar for parsing a sentence test suite illustrating most rules of French grammar. It is important that the suite includes not only positive examples but also negative examples to test the overgeneration of the grammar.

There are not many corpora of this type for French. We have chosen the TSNLP [3], which includes 1690 positive sentences and 1935 negative sentences. It is far from covering all of French grammar; in particular, it includes very few complex sentences but it stresses some phenomena such as coordination or the position in the sentence of the adverbial complements. On the other hand, our grammar covers phenomena that are ignored by the TSNLP: the passive and middle voice of verbs, the subcategorisation of predicative nouns and adjectives, the control of the subject of infinitive complements, the relative and interrogative clauses. . .

For the parsing, we used LEOPAR<sup>5</sup>, which is a parser devoted to IG. With the current grammar, the parser accepts 88% of the 1690 positive TSNLP sentences and rejects 85% of the 1935 negative sentences. The 15% of accepted negative sentences are due to the fact that the grammar ignores phonological rules and semantics. The 12% of unanalysed positive sentences are due to various reasons: speech sentences, frozen or semi-frozen expressions, phenomena that are not yet taken into account (causatives, superlatives. . .).

## 6 Prospects

The next step is to use our French grammar to parse raw corpora. It is already possible to use LEOPAR

with a large lexicon for such a task. It is necessary to enrich the grammar because some common linguistic phenomena are not yet taken into account. We also need to improve the efficiency of the parser to contain the possible explosion resulting from the increase of the grammar size in combination with the increased sentence length.

## References

- [1] J. Bresnan. *Lexical-Functional Syntax*. Blackwell Publishers, Oxford, 2001.
- [2] D. Duchier, J. Le Roux, and Y. Parmentier. XMG : Un compilateur de méta-grammaires extensible. In *TALN 2005, Dourdan, France*, 2005.
- [3] S. Lehmann, S. Oepen, S. Regnier-Prost, K. Netter, V. Lux, J. Klein, K. Falkedal, F. Fouvry, D. Estival, E. Dauphin, H. Compagnion, J. Baur, L. Balkan, and D. Arnold. TSNLP — Test Suites for Natural Language Processing. In *Proceedings of COLING 1996, Copenhagen*, 1996.
- [4] A. Nasr. A formalism and a parser for lexicalised dependency grammars. In *4th International Workshop on Parsing Technologies (IWPT)*, 1995.
- [5] G. Perrier. Interaction grammars. In *CoLing '2000, Sarrebrücken*, pages 600–606, 2000.
- [6] G. Perrier. La sémantique dans les grammaires d'interaction. *Traitement Automatique des Langues*, 45(3):123–144, 2004.
- [7] G. K. Pullum and B. C. Scholz. On the Distinction between Model-Theoretic and Generative-Enumerative Syntactic Frameworks. In *LACL 2001, Le Croisic, France*, volume 2009 of *Lecture Notes in Computer Science*, pages 17–43, 2001.
- [8] O. Rambow, K. Vijay-Shanker, and D. Weir. D-tree substitution grammars. *Computational Linguistics*, 27(1):87–121, 2001.
- [9] C. Retoré. *The Logic of Categorical Grammars*, 2000. *ESSLI'2000, Birmingham*.
- [10] J. Rogers and K. Vijay-Shanker. Obtaining trees from their descriptions: an application to tree-adjoining grammars. *Computational Intelligence*, 10(4):401–421, 1994.
- [11] I. A. Sag, T. Wasow, and E. M. Bender. *Syntactic Theory: a Formal Introduction*. Center for the Study of Language and INF, 2003.

<sup>5</sup> <http://www.loria.fr/equipes/calligramme/leopar>