# Parsing Ambiguous Structures using Controlled Disjunctions and Unary Quasi-Trees

**Philippe Blache**
LPL - CNRS
29 Avenue Robert Schuman
F-13621 Aix-en-Provence
pb@lpl.univ-aix.fr

## Abstract

The problem of parsing ambiguous structures concerns (*i*) their representation and (*ii*) the specification of mechanisms allowing to delay and control their evaluation. We first propose to use a particular kind of disjunctions called *controlled disjunctions*: these formulae allows the representation and the implementation of specific constraints that can occur between ambiguous values. But an efficient control of ambiguous structures also has to take into account lexical as well as syntactic information concerning this object. We then propose the use of *unary quasi-trees* specifying constraints at these different levels. The two devices allow an efficient implementation of the control of the ambiguity. Moreover, they are independent from a particular formalism and can be used whatever the linguistic theory.

## 1 Introduction

Most of the approaches dealing with ambiguity are disambiguating techniques. This preliminary constatation seems trivial and relies on a simple presuposition: the ambiguous structures need to be disambiguated. However, this is not true from several respects. Machine translation is a good example: the ambiguity of a sentence in the source language needs very often to be preserved and translated into the target one (cf. (Wedekind97)).

Another remark, in the same perspective: most of the disambiguating techniques rely on a single linguistic level. In other words, they generally make use of lexical or syntactic or semantic information, exclusively. But a natural processing of natural language should not work in this way. All the linguistic levels of NLP (i.e. phonetic, phonologic, lexical, syntactic, semantic and pragmatic) have to be taken into account *at the same time*. In other words, processing ambiguity would have to be parallel, not sequential. The problem is then to use ambiguous structures during the parse without blocking the analysis. In a first approximation, such a problem comes to parse using underspecified structures. We will see that this constitutes a part of the solution.

The third and last preliminary remark focuses on the control strategies for the evaluation of ambiguous structures. These strategies can rely on the formal properties of the ambiguous structure (for example the simplification of a disjunctive formula), on the contextual relations, etc. But the ambiguous objects can themselves bear important information specifying some restrictions. We will develop in this paper several examples illustrating this point. The approach described here make an intensive use of this kind of constraints, also called *control relations*.

We present in this paper a technique called *controlled disjunctions* allowing to represent and implement an efficient control of ambiguous structures at the lexical and phrase-structure level. We illustrate this technique using the HPSG framework, but it could be used in all kind of feature-based representations. This approach relies (*i*) on the representation of constraints relations between the feature values and (*ii*) on the propagation of such relations. We insist on the fact that this is not a disambiguating technique, but a control of the evaluation of ambiguous structures. In order to increase the number of constraints controlling an ambiguous structure, we generalize the use of control re-

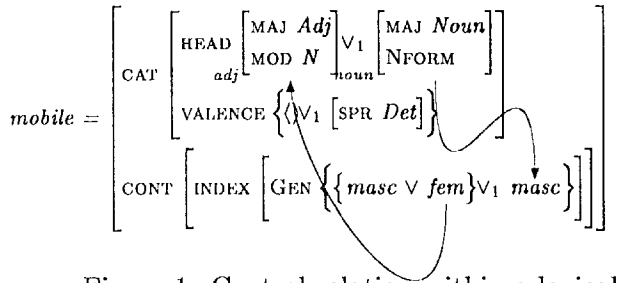



Figure 1: Control relation within a lexical entry

lations at the phrase-structure level. We propose for that a particular representation of hierarchical relations for ambiguous objects called *unary quasi-trees*.

This paper is threefold. In a first section, we present the limits of the classical representation of ambiguity and in particular the technique of named disjunctions. The second section describes the controlled disjunction method applied to the lexical level. We describe in the third section the generalization of this technique to the phrase-structure level using unary quasi-trees and we show how this approach is useful for an online control of the ambiguity during the parse.

## 2 Ambiguity and Disjunctions

Several techniques have been proposed for the interpretation and the control of disjunctive structures. For example, delaying the evaluation of the disjunctive formulae until obtaining enough information allows partial disambiguation (cf. (Karttunen84)). Another solution consists in converting the disjunctive formulae into a conjunctive form (using negation) as proposed by (Nakazawa88) or (Maxwell91). We can also make use of the properties of the formula in order to eliminate inconsistencies. This approach, described in (Maxwell91), relies on the conversion of the original disjunctive formulae into a set of *contexted constraints* which allows, by the introduction of propositional variables (*i*) to convert the formulae into a conjunctive form, and (*ii*) to isolate a subset of formulae, the *disjunctive residue* (the negation of the unsatisfiable constraints). The problem of the satisfiability of the initial formula is then reduced to that of the disjunctive residue.

This approach is fruitful and several methods rely on this idea to refer formulae with an index (a propositional variable, an integer, etc.). It is the case in particular with *named disjunctions* (see (Dörre90), (Krieger93) or (Gerdemann95)) which propose a compact representation of control phenomena and covariancy.

A named disjunction (noted hereafter ND) binds several disjunctive formulae with an index (the name of the disjunction). These formulae have the same arity and their disjuncts are ordered. They are linked by a covariancy relation: when one disjunct in a ND is selected (i.e. interpreted to true), then all the disjuncts occurring at the same position into the other formulae of the ND also have to be true. The example (1) presents the lexical entry of the german determiner *den*. The covariation is indicated by three disjunctive formulae composing the named disjunction indexed by 1.

(1)
$$den = \left[\begin{array}{l}\text{SPEC}\left[\begin{array}{l}\text{CASE}\left\{acc \vee_1 dat\right\} \\ \text{INDEX}\left[\begin{array}{l}\text{GEN}\left\{masc \vee_1 \_\right\} \\ \text{NUM}\left\{sing \vee_1 plu\right\}\end{array}\right]\end{array}\right]\end{array}\right]$$

But the named disjunction technique also has some limits. In particular, NDs have to represent all the relations between formulae in a covariant way. This leads to a lot of redundancy and a loss of the compactness in the sense that the disjuncts don't contain anymore the possible values but all the possible variancies according to the other formulae.

Some techniques has been proposed in order to eliminate this drawback and in particular: the *dependency group representation* (see (Griffith96)) and the *controlled disjunctions* (see (Blache97)). The former relies on an enrichment of the Maxwell and Kaplan's contexted constraints. In this approach, constraints are composed of the conjunction of base constraints (corresponding to the initial disjunctive form) plus a control formula representing the way in which values are choosen. The second approach, described in the next section, consists in a specific representation of control relations relying on a clear distinction between (*i*) the possible values (the disjuncts) and (*ii*) the relations between these ambiguous values and other elements of the structure. This approach allows a direct implementation of the implication relations (i.e. the oriented controls) instead of simple covariancies.

## 3 Controlled Disjunctions

The controlled disjunctions (noted hereafter CD) implement the relations existing between ambiguous feature values. The example of the figure (1) describes a non covariant relation between GENDER and HEAD features. More precisely, this relation is oriented: if the object is a noun, then the gender is masculine and if the object is feminine, then it is an adjective.

The relation between these values can be represented as implications: $noun \Rightarrow masc$ and $fem \Rightarrow adj$. The main interest of CDs is the representation of the variancy between the possible values and the control of this variancy by complex formulae.

Controlled disjunctions reference the formulae with names and all the formula are ordered. So, we can refer directly to one of the disjuncts (or to a set of linked disjuncts) with the name of the disjunction and its rank.

For clarity, we represent, as in the figure (2), the consequent of the implication with a pair indexing the antecedent. This pair indicates the name of the disjunction and the rank of the disjunct. In this example, $noun_{\langle 2,1\rangle}$ implements $noun \Rightarrow masc$: the pair $\langle 2,1\rangle$ references the element of the disjunction number 2 at the $1^{st}$ position.

(2)
$$mobile = \left[\begin{array}{l} \text{CAT} \left[\begin{array}{l} \text{HEAD } \boxed{1}\left\{ noun_{\langle \boxed{2},1\rangle},\ adj \right\} \\ \text{VALENCE} \mid \text{SPR } \left[\boxed{1}\left\{ [Det],\ [] \right\}\right] \end{array}\right] \\ \text{INDEX } \left[\text{GEN } \boxed{2}\left\{ masc,\ fem_{\langle \boxed{1},2\rangle} \right\}\right] \end{array}\right]$$

As shown in this example, CDs can represent covariant disjunction (e.g. the disjunction number 1) or simple disjunctions (disjunction number 2).

(3)
$$\left[\begin{array}{l} \text{U} = \left\{ a \vee_i a \vee_i a \vee_i b \vee_i b \vee_i b \right\} \\ \text{V} = \left\{ c \vee_i d \vee_i d \vee_i c \vee_i c \vee_i d \right\} \\ \text{W} = \left\{ f \vee_i e \vee_i f \vee_i e \vee_i f \vee_i e \right\} \end{array}\right]$$

The example (3)[1] presents the case of an ambiguity that cannot be totally controlled by a ND. This structure indicates a set of variancies. But the covariancy representation only implements a part of the relations. In fact, several "complex" implications (i.e. with a conjunction as antecedent) control these formulae as follows :

$$\{a \wedge c \Rightarrow f, b \wedge d \Rightarrow e, c \wedge e \Rightarrow b, d \wedge f \Rightarrow a\}$$

These implications (the "controlling formulae") are constraints on the positions of the disjuncts in the CD. The formula in the example (4) presents a solution using CDs and totally implementing all the relations. In this representation, $(i = 1) \wedge (j = 1) \Rightarrow (k = 2)$ implements the implication $a \wedge c \Rightarrow f$. The set of constraints is indicated into brackets. The feature structure, constrained by this set, simply contains the elementary variations.

(4)
$$\left\{\begin{array}{l} (i = 1) \wedge (j = 1) \Rightarrow (k = 2) \\ (i = 2) \wedge (j = 2) \Rightarrow (k = 1) \\ (j = 1) \wedge (k = 1) \Rightarrow (i = 2) \\ (j = 2) \wedge (k = 2) \Rightarrow (i = 1) \end{array}\right\} \rightarrow \left[\begin{array}{l} \left\{ a \vee_i b \right\} \\ \left\{ c \vee_j d \right\} \\ \left\{ e \vee_k f \right\} \end{array}\right]$$

From an implementation point of view, the controlled disjunctions can easily be implemented with languages using delaying devices. An implementation using functions in Life has been described in (Blache97).

[1] This problem was given by John Griffith.

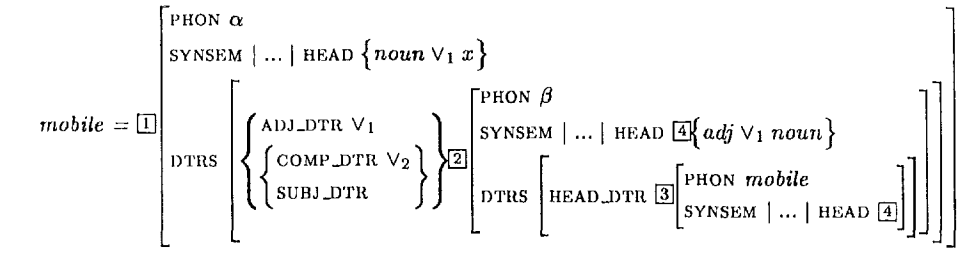



Figure 2: UQT in a HPSG form

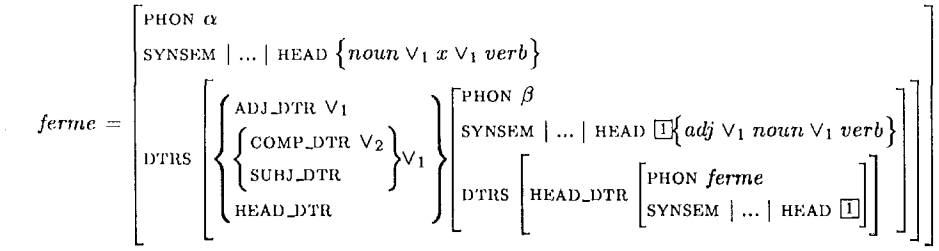



Figure 3: UQT of the lexical entry *ferme*

## 4 Generalization to the Phrase-Structure Level

### 4.1 Unary Quasi-Trees

(Vijay-Shanker92) proposes the use of trees description called *quasi-trees* whithin the framework of TAG. Such structures rely on the generalization of hierarchical relations between constituents. These trees bear some particular nodes, called quasi-nodes, which are constituted by a pair of categories of the same type. These categories can refer or not to the same objet. If not, a subtree will be inserted between them in the final structure.

Such an approach is particularly interesting for the description of generalizations. The basic principle in TAG consists in preparing subtrees which are part of the final syntactic structure. These subtrees can be of a level greater than one: in this case, the tree predicts the hierarchical relations between a category and its ancestors. Quasi-trees generalize this approach using a meta-level representation allowing the description of the general shape of the final syntactic tree.

The idea of the *unary quasi-trees* relies basically on the same generalization and we propose to indicate at the lexical level some generalities about the syntactic relations. At the difference with the quasi-trees, the only kind of information represented here concerns hierarchy. No other information like subcategorization is present there. This explain the fact that we use *unary* trees.

Several properties characterizes unary quasi-trees (noted hereafter UQTs):

- An UQT is interpreted from the leaf (the lexical level) to the root (the propositional one).
- A relation between two nodes $\alpha$ and $\beta$ ($\alpha$ dominating $\beta$) indicates, in a simple PSG representation, that there exists a derivation of the form $\alpha \Rightarrow^* B$ such that $\beta \in B$.
- Each node has only one daughter.
- An unary quasi-tree is a description of tree and each node can be substituted by a subtree[2].

[2]But at the difference with the quasi-trees, a node is not represented by a pair and no distinction is done between *quasi-root* and *quasi-foot* (see (Vijay-Shanker92)).

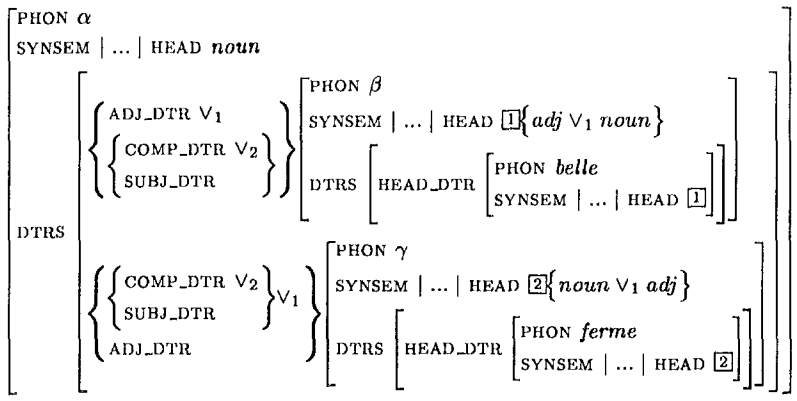
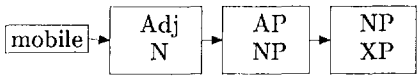



Figure 4: UQT with an embedded ambiguity

- The nodes can be constituted by a set of objects[3]. If more than one object compose a node, this set in interpreted as a disjunction. Such nodes are called ambiguous nodes. A categorial ambiguity is then represented by an unary quasi-tree in which each node is a set of objects.
- Each node is a disjunctive formula belonging to a covariant disjunction.
- An UQT is limited to three levels: lexical, phrase-structure and propositional.

(5)




The example (5) shows the UQT corresponding to the word *mobile* with an ambiguity adjective/noun. For clarity's sake, the tree is presented upside-down, with the leaf at the top and the root at the bottom. This example indicates that:

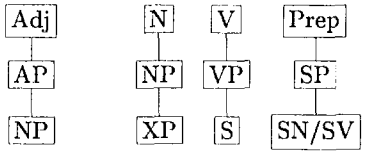
- an adjective is a daughter of an AP which is to its turn a daughter of a NP,
- a noun is a daughter of a NP which is to its turn a daughter of an unspecified phrase XP.

[3]These objects, as for the quasi-trees, can be constituted by atomic symbols or feature structures, according to the linguistic formalism.

As indicated before, each node represents a disjunctive formula and the set of nodes constitutes a covariant disjunction. This information being systematic, it becomes implicit in the representation of the UQTs (i.e. no names are indicated). So, the position of a value into a node is relevant and indicates the related values into the tree.

This kind of representation can be systematized to the major categories and we can propose a set of elementary hierarchies, as shown in the figure (6) used to construct the UQTs.

(6)




It is interesting to note that the notion of UQT can have a representation into different formalisms, even not based on a tree representation. The figure (2) shows for example an HPSG implementation of the UQT described in the figure (1).

In this example, we can see that the ambiguity is not systematically propagated to all the levels: at the second level (substructure 2), both values belong to a same feature (HEAD-DAUGHTER). The covariation here concerns different features at different levels. There is for example a covariation between the HEAD features of the second level and the

type of the daughter at the third level. Moreover, we can see that the noun can be projected into a NP, but this NP can be either a complement or a subject daughter. This ambiguity is represented by an embedded variation (in this case a simple disjunction).

The example described in the figure (3) shows a french lexical item that can be categorized as an adjective, a noun or a verb (resp. translated as *ferm*, *farm* or *to close*). In comparison with the previous example, adding the verb subcase simply consists in adding the corresponding basic tree to the structure. In this case, the covariant part of the structure has three subcases.

This kind of representation can be considered as a description in the sense that it works as a constraint on the corresponding syntactic structure.

### 4.2 Using UQTs

The UQTs represent the ambiguities at the phrase-structure level. Such a representation has several interests. We focus in this section more particularly on the factorization and the representation of different kind of constraints in order to control the parsing process.

The example of the figure (4) presents an ambiguity which "disappears" at the third level of the UQT. This (uncomplete) NP contains two elements with a classical ambiguity *adj/noun*. In this case, both combinations are possible, but the root type is always *nominal*. This is an example of ambiguous structure that doesn't need to be disambiguated (at least at the syntactic level): the parser can use directly this structure[4].

As seen before, the controlled disjunctions can represent very precisely different kind of relations within a structure. Applying this technique to the UQTs allows the representation of dynamic relations relying on the context. Such constraints use the selection relations existing between two categories. In case of ambiguity, they can be applied to an ambiguous group in order to eliminate inconsistencies and control the parsing process. In this case, the goal is not to disambiguate the structure, but (*i*) to delay the evaluation and maintain the ambiguity and (*ii*) in order to reduce the set of solutions. The figure (5) shows an example of the application of this technique.

The selection constraints are applied between some values of the UQTs. These relations are represented by arcs between the nodes at the lexical level. They indicate the possibility of cooccurrence of two juxtaposed categories. The constraints represented by arrows indicate subcategorization. If such constraint is applied to an ambiguous area, then it can be propagated using the selection constraints whithin this area. In this example, there is a selection relation between the root *S* of the UQT describing *"possède"* and the node value *NP* at the second level of the UQT describing *"ferme"*. This information is propagated to the rest of the UQT and then to the previous element using the relation existing between the values *N* of *"ferme"* and *Adj* of *"belle"*. All these constraints are represented using controlled disjunctions: each controller value bears the references of the controlled one as described in the section (3).

The interest of this kind of constraints is that they constitute a local network which defines in some way a controlled ambiguous area. The parsing process itself can generate new selection constraints to be applied to an entire area (for example the selection of a NP by a verb). In this case, this constraint can be propagated through the network and eliminate inconsistent solutions (and eventually totally disambiguate the structure). This pre-parsing strategy relies on a kind of head-corner method. But the main goal here, as for the lexical level, is to provide constraints controlling the disambiguation of the structures, not a complete parsing strategy.

## 5 Conclusion

*Controlled Disjunctions* allow a precise representation of the relations occuring between feature values. Such relations can be defined

[4]We can also notice that covariation implements the relation between the categories in order to inhibit the *noun/noun* or *adj/adj* possibilities (cf. the CD number 1).

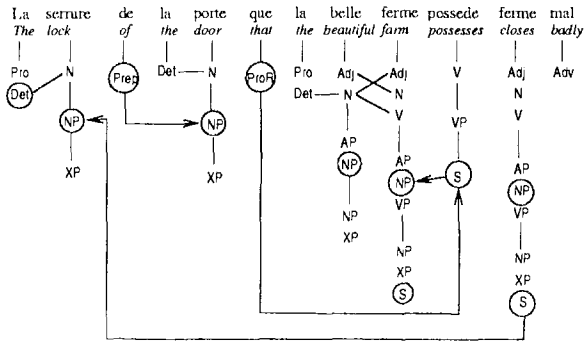



Figure 5: Constraint networks on ambiguous areas

statically, in the lexicon. They can also be introduced dynamically during the parse using the *Unary Quasi-Tree* representation which allows the description of relations between categories together with their propagation. These relations can be seen as constraints used to control the parsing process in case of ambiguity.

An efficient treatment of the ambiguity relies on the possibility of delaying the evaluation of ambiguous structures (i.e. delaying the expansion into a disjunctive normal form). But such a treatment is efficient if we can (1) extract as much information as possible from the context and (2) continue the parse using ambigous structures. The use of CDs and UQTs constitutes an efficient solution to this problem.

## References


Philippe Blache. 1997. "Disambiguating with Controlled Disjunctions." In *Proceedings of the International Workshop on Parsing Technologies.*

Jochen Dörre & Andreas Eisele. 1990. "Feature Logic with Disjunctive Unification" in proceedings of *COLING'90.*

Dale Gerdemann. 1995. "Term Encoding of Typed Feature Structures." In *Proceedings of the Fourth International Workshop on Parsing Technologies*, pp. 89–98.

John Griffith. 1996. "Modularizing Contexted Constraints." In *Proceedings of COLING'96.*

Lauri Karttunen. 1984. "Features and Values" in proceedings of *COLING'84.*

Robert Kasper & William Rounds 1990. "The Logic of Unification in Grammar" in *Linguistics and Philosophy*, 13:1.

Hans-Ulrich Krieger & John Nerbonne. 1993. "Feature-Based Inheritance Networks for Computational Lexicons." In T. Briscoe, V. de Paiva and A. Copestake, editors, *Inheritance, Defaults and the Lexicon.* Cambridge University Press, Cambridge, USA.

John T. Maxwell III & Ronald M. Kaplan. 1991. "A Method for Disjunctive Constraints Satisfaction." In M. Tomita, editor, *Current Issues in Parsing Technology.* Kluwer Academic Publishers, Norwell, USA.

Tsuneko Nakazawa, Laura Neher & Erhard Hinrichs. 1988. "Unification with Disjunctive and Negative Values for GPSG Grammars" in proceedings of *ECAI'88.*

Gertjan van Noord & Gosse Bouma. 1994 "Adjuncts and the Processing of Lexical Rules" in proceedings of *COLING'94.*

K. Vijay-Shanker. 1992 "Using Descriptions of Trees in a Tree Adjoining Grammar" in *Computational Linguistics*, 18:4.

Jürgen Wedekind & Ronald Kaplan. 1997 "Ambiguity-Preserving Generation with LFG- and PATR-style Grammars" in *Computational Linguistics*, 22:4.