PROPERTY GRAMMARS: A SOLUTION FOR PARSING WITH CONSTRAINTS

Philippe Blache
LPL – Université de Provence, 29 Av. Robert Schuman, 13621 Aix-en-Provence, France
pb@lpl.univ-aix.fr

1. Situation: Guidelines for an Actual Constraint-Based Approach

Modern linguistic theories often make use of the notion of *constraint* to represent information. This notion allows a fine-grained representation of information, a clear distinction between linguistic objects and their properties, and a better declarativity. Several works try to take advantage of a constraint-based implementation (see for example [Maruyama90], [Carpenter95], [Duchier99]). However, the parsing process cannot be interpreted as an actual constraint satisfaction one. This problem mainly comes from the generative conception of grammars on the linguistic analysis. Indeed, in constraint-based theories, constraints can appear at different levels: lexical entries, grammar rules, universal principles. However, during a parse, one has first to select a local tree and then to verify that this tree satisfies the different contraints. This problem comes from the generative interpretation of the relation between grammars and languages. In this case, the notion of derivation is central and parsing consists in finding a derivation generating an input. We propose then a new formalism called *Property Grammars* representing the linguistic information by means of constraints. These constraints constituting an actual system, it becomes possible to consider parsing as a satisfaction process.

An optimal use of constraints should follow some requirements. In particular, all linguistic information has to be represented by means of constraints. This information constitutes a system of constraints, then all the constraints are at the same level and the order of verification of the constraints is not relevant. Encapsulation is another important characteristics which stipulates that a constraint must represent homogeneous information.

The last important point concerns the notion of grammaticality. In the particular problem of parsing, finding an exact solution consists in associating a syntactic structure to a given input. In the case of generative approaches, this amounts to finding a derivation from a distinguished symbol to this input. However, the question when parsing real natural language inputs should not be the grammaticality, but the possibility of providing some information about the input. We propose then to replace the notion of grammaticality with that of *characterization* which is much more general: a characterization is the state of the constraint system for a given input.

2. Property Grammars

Property Grammars (cf. [Blache99]) provide a framework implementing these requirements: declarativity, encapsulation, satisfiability. We use for the representation of syntactic information 7 properties defined as follows :

• **Constituency** (noted *const*)	Set of categories constituting a phrase.
• **Obligation** (noted *oblig*)	Set of compulsory, unique categories (heads).
• **Unicity** (noted *uniq*)	Set of categories which cannot be repeated in a phrase.
• **Requirement** (noted $\Rightarrow$)	Cooccurrency between sets of categories.
• **Exclusion** (noted $\not\Leftrightarrow$)	Restriction of cooccurrence between sets of categories.
• **Linearity** (noted $\prec$)	Linear precedence constraints.
• **Dependency** (noted $\leadsto$)	Dependency relations between categories.

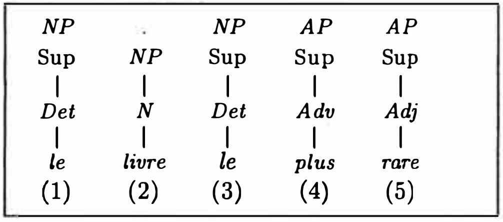
It is interesting to notice that properties can be expressed over sets of categories, allowing then to represent contextual information.

3. Parsing as Constraint Satisfaction

The parsing process (1) takes as input the set of elementary trees (in fact *unary quai trees*, as described in [Blache98]) that can be associated to the sentence and (2) builds all the characterizations of this input. A *characterization* is then defined over a set of categories by P^+ (the set of satisfied properties) and P^-(the set of unsatisfied properties). The following example describes a grammar of the NP in french.

Properties of the NP: (1) $Const = \{Det, N, AP, Sup\}$ (2) $Oblig = \{N, AP\}$ (3) $N[com] \Rightarrow Det$ (4) $Det \prec N$ (5) $Det \prec AP$ (6) $Det \prec Sup$ (8) $N \prec Sup$ (9) $AP \not\Leftrightarrow Sup$ (10) $Det \leadsto N$ (11) $AP \leadsto N$ (12) $Sup \leadsto N$
Properties of the AP: (1) $Const = \{Adj, Adv\}$ (2) $Oblig = \{Adj\}$ (3) $Adv \prec Adj$ (4) $Adv \leadsto Adj$
Properties of the Sup (superlative): (1) $Const = \{Det, Adv, Adj\}$ (2) $Oblig = \{Adj\}$ (3) $Adj \Rightarrow Det$ (4) $Adj \Rightarrow Adv$ (5) $Det \prec Adv$ (6) $Det \prec Adv$ (7) $Adv \prec Adj$ (8) $Det \leadsto Adj$ (9) $Adv \leadsto Adj$

The following example presents the elementary trees associated to the words of the noun phrase *le livre le plus rare (the rarest book).*



The general approach consists in characterizing all subsets of categories belonging to the set of elementary trees, whatever their level. Considering all the possible subsets of categories has an exponential cost and some simple controls can be applied. In the following, we will only take into account the sets of juxtaposed categories. Building the characterizations consists for each subset of categories of verifying the satisfiability of the properties.

The following example illustrates the construction of such characterizations for the grammar presented above. The properties are represented by their indexes. For example, the characterization of the set $\{Det_1/N_2\}$ indicates that all the properties specified for these categories (i.e. constituency, obligation, requirement, precedence and dependency) are satisfied. The set $\{Adv_4/AP_5\}$ presents two characterizations: this sequence can characterize an *AP* (all the properties of the grammar are verified) and a *Sup* in which a requirement property is not satisfied.

Characterization of 2-uples:	- Det_1/N_2 :	$P^+(NP) = \{1, 2, 3, 4, 10\}$;	$P^-(NP) = \emptyset$
	- Adv_4/AP_5 :	$P^+(AP) = \{1, 2, 3, 4\}$;	$P^-(AP) = \emptyset$
		$P^+(Sup) = \{1, 2, 4, 9\}$;	$P^-(Sup) = \{3\}$
	...		
Characterization of 3-uples:	- $Det_1/N_2/Sup_3$:	$P^+(NP) = \{1, 2, 3, 4, 6, 8, 9, 10, 12\}$;	$P^-(NP) = \emptyset$
	- $Sup_1/N_2/Det_3$:	$P^+(NP) = \{1, 2, 3, 9, 10, 12\}$;	$P^-(NP) = \{4, 6, 8\}$
	...		
Characterization of 4-uples	- $Det_1/N_2/Sup_3/AP_4$:	$P^+(NP) = \{1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12\}$;	$P^-(NP) = \{9\}$
	...		

Characterizations are associated to a given set of categories without taking into consideration the characterization of each of these categories. However, the information contained by the set of characterizations for a given input allow the reconstitution of a hierarchical structure. The mechanism consists in finding, if possible, a characterization covering the input, then finding characterizations for the embedded constituents. The following example describes such a process. For clarity, only positive characterizations are examined here.

Among the characterizations given in the figure (2), 4 are positive:

- Det_1/N_2 :	$P^+(NP) = \{1, 2, 3, 4, 10\}$;	$P^-(NP) = \emptyset$
- Adv_4/Adj_5 :	$P^+(AP) = \{1, 2, 3, 4\}$;	$P^-(AP) = \emptyset$
- $Det_1/N_2/Sup_3$:	$P^+(NP) = \{1, 2, 3, 4, 8, 9, 10, 12\}$;	$P^-(NP) = \emptyset$
- $Det_3/Adv_4/Adj_5$:	$P^+(Sup) = \{1, 2, 3, 4, 5, 6, 8, 9\}$;	$P^-(NP) = \emptyset$

There is in this case only one solution covering all the input (using only positive characterization): the set of categories $Det_1/N_2/Sup_3$ and $Det_3/Adv_4/Adj_5$ which characterize respectively a *NP* and a *Sup*.

A naive implantation of these mechanisms is obviously not efficient. Indeed, let m be the number of words of the input, c the maximum number of categorizations for each word, then the number n of categories to analyze is bounded by $n = 2mc$. Then, the number of sets of categories to analyse is bounded by 2^n.

Several controls can be added in order (1) to reduce the number of sets to analyze and (2) to control the satisfaction process. We can for example choose to build only sets of juxtaposed categories or consider only sets actually characterized by properties. The satisfaction process itself can be controlled using several heuristics: it is possible to filter the satisfiability according to a threshold given by the cardinality of the set of unsatisfied properties P^-: it is possible to build only characterizations with less than a certain amount of unsatisfied constraints. At the extreme, we can reduce the satisfiability to positive characterizations (reducing characterization to grammaticality).

4. Conclusion

The representation of linguistic information by means of constraints is interesting both for knowledge representation (no implicit information, encapsulation) and for implementation point of view (parsing is seen as a constraint satisfaction process). Property Grammars offer a framework taking advantage of these characteristics and present several interests for natural language processing. The first one concerns the generality and the robustness of the approach: constraints allow the introduction of the notion of characterization to replace that of grammaticality. The parsing process consists in verifying constraints for the possible set of categories associated to the input instead of trying to build a structure according to the grammar. One other interesting point concerns the integration of different information sources: the properties can concern all kind of information (prosodic, syntactic, pragmatiç etc.).

References

[Archangeli97] Archangeli D. & D.T. Langendoen eds. (1997) *Optimality Theory*, Blackwell.

[Blache98] Blache P. (1998) "Parsing Ambiguous Structures using Controlled Disjunctions and Unary Quasi-Trees." in proceedings of *ACL-COLING'98.*

[Blache99] Blache P. (1999) *Filtering and Fusion: A Technique for Parsing with Properties*, in proceedings of NLPRS'99.

[Carpenter95] Carpenter B. & Penn G. (1995) "Compiling Typed Attribute-Value Logic Grammars", in H. Bunt and M. Tomita (eds.), *Current Issues in Parsing Technologies*, Kluwer.

[Duchier99] Duchier D. & Thater S. (1999) "Parsing with Tree Descriptions: a constraint based approach", in proceedings of NLULP'99.

[Maruyama90] Maruyama H. (1990), "Structural Disambiguation with Constraint Propagation", in proceedings of *ACL'90.*

[Shieber92] Shieber S. (1992) *Constraint-Based Grammar Formalisms*, MIT Press.