"*Model Theoretic Syntax is not Generative Enumerative Syntax with constraints*": under what condition?

Philippe Blache

Laboratoire Parole et Langage
Aix-Marseille Universités & CNRS
`pb@lpl.univ-aix.fr`

Abstract. The generative conception of grammars relies on the derivation process which, to its turn, depends on a hierarchical representation of syntax. We show in this paper how a fully constraint-based approach, avoiding such restriction, can constitute an alternative to generativity and form the basis of a framework for a model-theoretic conception of grammar.

1 Introduction

There are two different approaches in logic: one is *syntactic* and only uses the form of the formulae in order to demonstrate a theorem, the second is *semantic* and relies on formulae interpretation. The same distinction also holds in linguistics. A first conception relies on the study of the input formedness. In this case, the problem consists in finding a structure adequate to the input. Grammatical information is then represented by means of a set a rules. An alternative conception, instead of relying on the input form, focuses on its characteristics. Following [Pullum & Scholz 01], we will call these approaches respectively generative enumerative syntax (GES) and model-theoretic syntax (MTS) The first approach is that of generative theories, it has been extensively experimented. The latter still remains marginal. One of the reasons is that generativity has been for years almost the unique view for formal syntax and it is difficult to move from this conception to a different one. In particular, one of the problem comes from the fact that all approaches, even those in the second perspective, still rely on a hierarchical (tree-like) representation of syntactic information.

This paper describes this problem and proposes an MTS framework moving from a classical tree domain towards a graph domain for the representation of syntactic information. We show how constraints can be an answer to this problem: first, they can represent all kinds of syntactic information and second, they constitute a system, all constraints being at the same level (none being to be evaluated before others).

The paper starts with a description of the main characteristics of these different conceptions of syntax. In the second section, we focus on the specific problems coming from the hierarchical conception of syntax, showing how it can constitute a severe limitation for linguistic description. The third section proposes an overview of an MTS framework, called *Property Grammars*, following this requirements. We precise formally the status of the constraints we use, and how in this approach a syntactic description comes to a graph. We explain in particular how it is possible to take advantage of such a representation in order to shift from the classical tree domain to a graph one, and in what sense this shift constitutes a solution to the MTS problem.

2 Proof vs. Model Theory in Syntax

The generative conception of syntax relies on a particular relation between grammar and language: a specific mechanism, derivation, makes it possible to generate a language from a grammar. This basic mechanism can be completed with other devices (transformations, moves, feature propagation, etc.) but in all cases constitute the core of all generative approaches. In such case, grammaticality consists in finding a set of derivations between the start symbol of the grammar and the sentence to be parsed. As a side effect, a derivation step coming to a local tree, it is possible to build a syntactic structure, represented by a tree. It is then possible to reduce in a certain sense the question of grammaticality to the possibility of building a tree. This reminder seems to be trivial, but it is important to mesure its consequences. The first is that grammaticality is reduced, as it has been noticed in [Chomsky75], to a boolean value: true when a tree can be built, false otherwise. This is a very restrictive view of grammaticality, as it also has been noticed in [Chomsky75] (without proposing a solution), which forbids a finer conception, capable of representing in particular a grammaticality scale (also called *gradience*, see [Keller00] or [Aarts07]).

This generative conception of syntax is characterized as being enumerative (see [Pullum & Scholz 01] in the sense that derivation can be seen as an enumeration process, generating all possible structures and selecting them by means of extra constraints (as it is typically the case in the Optimality Theory, see [Prince93]).

Model Theoretic Syntax proposes an alternative view ([Blackburn & al. 93], [Cornell & Rogers 00], [Pullum & Scholz 01]). In this conception, a grammar is a set of assessments, the problem consists in finding a model into a domain.

From a logical perspective, generative approaches rely on a *syntactic* conception in the sense that parsing consists in applying rules depending on the form of the structures generated at each step. For example, a nonterminal is replaced with a set of constituents. On the opposite, model-theoretic approaches rely on a *semantic* view in which parsing is based on the interpretation (the truth values) of the statements of the grammar. A grammar in MTS is a set of statements or, formally speaking, formulae. Each formula describes a linguistic property; its interpretation consists in finding whether this statement is true or false for a given set of values (the universe of the theory in logical terms). When a set of values satisfies all assessments of the grammar (in other words when the interpretation of all the formulae for this set of values is true), then this set is said to be a model.

As far as syntax is concerned, formulae indicate relations between categories or, more precisely, between descriptions of categories. These descriptions correspond to the specification of a variable associated with several properties: they can be seen as formulae. For example, given $\mathcal{K}$ a set of categories, a description of a nominative noun comes to the formula:

(1) $\exists x[cat(x, N) \wedge gend(x, masc)]$

A category can be described by a more or less precise description, according to the number of conjuncts. A grammatical statement is a more complex formula, adding to the categories descriptions other relations. For example, a statement indicating that a determiner is unique within a noun phrase comes to the formula:

(2) $[cat(x, Det) \wedge cat(y, Det) \rightarrow x \approx y]$

Concretely, when parsing a given input, a set of categories is instantiated, making it possible to interpret all the atomic predicates corresponding to the features (category, gender, number, etc.), making it possible to interpret to their turn the complex predicates formed by the grammatical statements. In this perspective, we say that an instantiated category is a value and finding a model consists in finding a set of values satisfying all the grammatical statements. For example, the set of words "*the book*" makes it possible to instantiate two categories with labels *Det* and *N* (these labels representing the conjunction of features). Intuitively, we can say that the set of values *{Det, N}* is a model for the *NP*.

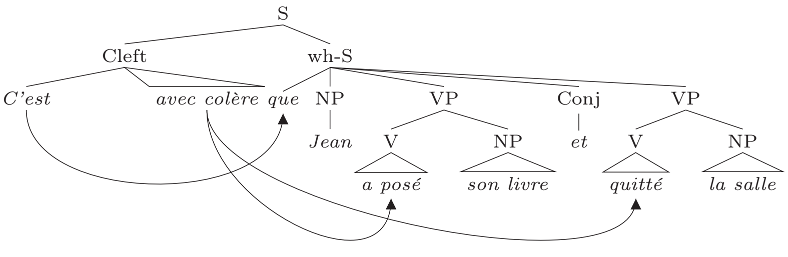
Finding a model is then completely different than deriving a structure. As underlined by [Pullum & Scholz 01], instead of enumerating a set of expressions, an MTS grammar simply states conditions on these expressions.

3 Generativity and hierarchical structures

Geoffrey Pullum, during a lecture at ESSLLI in 2003, explained that "*Model Theoretic Syntax is not Generative Enumerative Syntax with constraints*". In other words, constraints are not to be considered only as a control device (in the DCG sense for example) but have to be part of the theory. Some theories (in particular HPSG) try to integrate this aspect. But it remains an issue both for theoretical and technical reasons. The problem comes in particular from the fact usually, dominance relation plays a specific role in the representation of syntactic information: dominance structures have first to be built before verifying other kinds of constraints. This is a problem when no such hierarchical relations can be identified. Moreover, we know since GPSG that dominance constitutes only a part of syntactic information to be represented in phrase-structure approaches, not necessarily to be considered as more important than others.

Syntactic information is usually defined, especially in generative approaches, over tree domains. This is due to the central role played by the notion of dominance, and more precisely by the relation existing between the head and its direct ancestor. In theories like HPSG (see [Sag al. 03]), even though no rules are used (they are replaced with abstract schemata), this hierarchical organization remains at the core of the system. As a consequence, constraints in HPSG can be evaluated provided that a tree can be built: features can be propagated and categories can be instantiated only when the hierarchical skeleton is known. This means that one type of information, dominance, plays a specific role in the syntactic description.

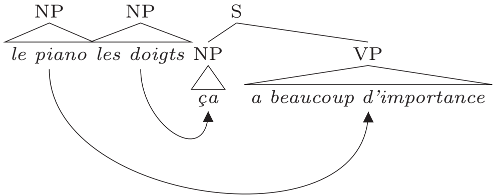
However, in many cases, a representation in terms of tree is not adapted or even not possible. The following example illustrate this situation. It present the case of a cleft element adjunct of two coordinated verbs.



C'est avec colère que Jean a posé son livre et quitté la salle
It is with anger that Jean put his book and left the room

Arrows in this figure shows in what sense the tree fails in representing the distribution of the cleft element onto the conjuncts. Moreover, there also exist other kinds of relations, for example the obligatory cooccurrence in French between "*c'est*" and "*que*".

The second example, presented in the following structure, illustrates the fact that in many cases, it is not possible to specify clearly what kind of syntactic relation exists between different parts of the structure:



Le piano les doigts ça a beaucoup d'importance
The piano the fingers it has a lot of immportance

This example illustrate a multiple detachment construction. In this case, detached element are not directly connected by classical syntactic relations to the rest of the structure: the two relations indicated by arrows are dependencies at the discourse level (plus an anaphoric relation).

Many other examples can be given, illustrating this problem: it is not always possible to give a connected structure on the basis of syntactic relations. Moreover, when adding other kinds of relations, the structure is not anymore a tree. This conception has direct consequences on the notion of grammaticalness. First, building a tree being a pre-requisite, nothing can be said about the input when this operation fails. This is the main problem with generative approaches that can only indicate whether or not an input is grammatical, but do not explain the existence of levels of grammaticality (the gradience phenomenon, see [Keller00], [Pullum & Scholz 01]).

A second consequence concerns the nature of linguistic information, that is typically spread over different domains (prosody, syntax, pragmatics, and related domains such as gestures, etc.). An input, in order to be interpreted, does not necessarily need to receive a complete syntactic structure. The interpretation rather consists in bringing together pieces of information coming from these different domains. This means that interpreting an input requires to take into account all the different domains and their interaction, rather than building a structure for each of them and then calculating their interface. In this perspective, no specific relation plays a more important role than others. This is also true within domains: as for syntax, the different properties presented in the previous section has to be evaluated independently from the others.

4 A constraint-based MTS framework: Property Grammars

A seminal idea in GPSG (see [Gazdar & al. 85]) was to dissociate the representation of different types of syntactic information: dominance and linear precedence (forming the ID/LP formalism), but also some other kinds of information stipulated in terms of cooccurrence restriction. This proposal is not only interesting in terms of syntactic knowledge representation (making it possible to factorize rules, for example), but also theoretically. Remind that one of the main differences between GES and MTS frameworks lies in the relation between grammar and language: MTS approaches try to characterize an input starting from available information, with no need to "overanalyze", to re-build (or infer) information that is not accessible from the input. For example, GES techniques have to build a connex and ordered structure, representing the generation of the input. On the opposite, nothing in MTS imposes to build a structure covering the input, which makes it possible for example to deal with partial or heterogeneous information. *Property Grammars* (see [Blache05]) systematizes the GPSG proposal in specifying these different types. More precisely, they propose to represent separately the following properties:

- *Constituency*: set of all the possible elements of a construction
- *Uniqueness*: constituents that cannot be repeated within a construction

- *Linearity*: linear order
- *Obligation*: set of obligatory constituents, one of them (exclusively to the others) being realized.
- *Requirement*: obligatory cooccurrence between constituents within a construction
- *Exclusion*: impossible cooccurrence between constituents within a construction

This list is not closed and other types of information can be added. For example, dependency (syntactico-semantic relation between a governor and a complement), or adjacency (juxtaposition of two elements). We focus in this paper on the 6 basic relations indicated above. These relations makes it possible to represent most of the syntactic information. We call these relations "properties", they can also be considered as constraints on the structure.

We adopt in the remaining of this paper the following notations: x, y (lower case) represent individual variables; X, Y (upper case) are set variables. We note $C(x)$ the set of individual variables in the domain assigned to the category C (see [Backofen & al. 95] for more precise definitions). We use the set of binary predicates for immediate domination ($\triangleleft$), linear precedence ($\prec$) and equality ($\approx$).

Let us now define more precisely the different properties. The first one (constituency) implements the classical immediate dominance relation. The others can be defined as follow:

- $Const(A, B) : (\forall x, y)[(A(x) \wedge B(y) \rightarrow x \triangleleft y]$
  This is the classical definition of constituency, represented by the dominance relation: a category B is constituent of A stipulates that there is a dominance relation between the corresponding nodes.
- $Uniq(A) : (\forall x, y)[A(x) \wedge A(y) \rightarrow x \approx y]$
  If one node of category A is realized, there cannot exists other nodes with the same category A. Uniqueness stipulates constituents that cannot be repeated in a given construction.
- $Prec(A, B) : (\forall x, y)[(A(x) \wedge B(y) \rightarrow y \not\prec x)]$
  This is the linear precedence relation as proposed in GPSG. If the nodes x and y are realized, then y cannot precedes x
- $Oblig(A) : (\exists x)(\forall y)[A(x) \wedge A(y) \rightarrow x \approx y]$
  There exists a node x of category A and there is no other node y of the same category. An obligatory category is realized exactly once.
- $Req(A, B) : (\forall x, y)[A(x) \rightarrow B(y)]$
  If a node x of category A is realized, a node y of category B has too. This relation implements cooccurrence, in the same way as GPSG does.
- $Excl(A, B) : (\forall x)(\nexists y)[A(x) \wedge B(y)]$
  When x exists, there cannot exist a sibling y. This is the exclusion relation between two constituents.

What is interesting in this representation of syntactic information is that all relations are represented independently form each others. They all are assessment in the MTS sense, and they can be evaluated separately (which fits well with the non-holistic view of grammatical information proposed by Pullum). In other words there is no need to assign the dominance relation a specific role: this is one information among others, what is meaningful is the interaction between these relations. More precisely, a set of categories can lead to a well-formed structure when all these assessments are satisfied, altogether. We do not need first to build a structure relying on dominance and then to verify other kind of information represented by the rest of the relations. In other words, in this approach, "*MTS is not GES with constraints*" ([Pullum & Scholz 03]).

Concretely, when taking into consideration a set of categories (an assignment), building the syntactic structure comes to evaluate the constraint system for this specific assignment. The result of the evaluation indicates whether or not the assignment corresponds to a well-formed list of constituents. For example, given two nodes x and y, if they only verify a precedence relation, nothing else can be said. But when several other properties such that requirement, uniqueness, constituency are also satisfied, the assignment $\{x, y\}$ becomes a model for an upper-level category. For example, if we have x and y such that $Det(x)$ and $N(y)$, this assignment verifies precedence, uniqueness, constituency and requirement properties. This set of properties makes it possible to characterize a *NP*. At the opposite, if we take x and y such that $Det(x)$ and $Adv(y)$, no constraint involving both constituents belong to the system: they do not constitue a model, and no new category can be infer.

In terms of representation, at the difference with classical approaches, syntactic information is not represented by means of a tree (see [Huddleston & Pullum 02]), but with a *directed labelled graph*. Nodes are categories and edges represent constraints over the categories: dominance, precedence, requirement, etc. A non-lexical category is described by a set of constraints, that are relations between its constituents. It is possible to take under consideration only one type of property (in other words one type of relation): this comes to extract a subgraph from the total one. For example, one can consider only constituency properties. In this case, the corresponding subgraph of dominance relations is (generally) a tree. But what is needed to describe precisely an input is the entire set of relations.

In the following, we represent the properties with the set of relations noted $\Rightarrow$ (requirement), $\otimes$ (exclusion), $\circ$ (uniqueness), $\triangleleft$ (constituency), $\uparrow$ (obligation), $\prec$ (precedence). A *Property Grammar* graph (noted *PG-graph*) is a tuple of the form:

$$G = \langle W, \Rightarrow, \otimes, \circ, \triangleleft, \uparrow, \prec, \theta \rangle$$

in which W is the set of nodes, θ the set of terminal nodes. A model is a pair $\langle G, V \rangle$ where V is a function from W to *Pow(W)*. We describe in the next section the use of such graphs.

5 Grammaticalness and constraints

Classically, syntactic information is usually represented in terms of decorated ordered trees (see [Blackburn & al. 93], [Blackburn & Meyer-Viol 94]). In this approach, tree admissibility relies on a distinction between dominance relation (that gives the structure) and other constraints on the tree such as precedence, cooccurrence restriction, etc. In our view, all relations has to be at the same level. In other words, dominance does not play a specific role: cooccurrence restriction for example can be expressed and evaluated independently from dominance. This means that each property represents a relation between nodes, dominance being one of them. When taking into consideration the entire set of relations, the structure is then a graph, not a tree. More precisely, each property specifies a set of relations between nodes: precedence relations, cooccurrence relations, dominance relations, etc. It can be the case that the dominance subset of relations (a subgraph of the graph of relations), is a tree, but this can be considered as a side effect. No constraint for example stipulates a connexity restriction on the dominance subgraph.

In *PG*, a grammar is then conceived as a constraint system, corresponding to a set of properties as defined above. Parsing an input consists in finding a model satisfying all the properties (or more precisely, the properties involving the categories of an assignment). In this case, the input is said to be grammatical, its description being the set of such properties. However, it is also possible to find models that satisfy partially the system. This means that some constraints can be violated. If so, the input is not grammatical, but the set of satisfied and violated properties still constitute a good description. We call such set a *characterization.* This notion replaces that of grammaticalness (which is a particular case of characterization in which no property is violated).

The following example (figure 1b) illustrates the case of an assignment A=*{NP, Det, Adj, N}*. All properties are satisfied, each relation forms an labeled edge, the set of relations being a graph. A phrase is characterized when it is connected to a graph of properties.

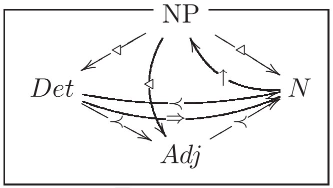


Fig. 1a

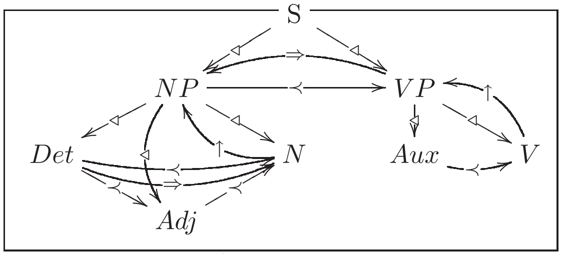
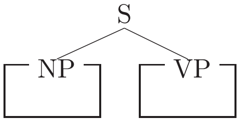
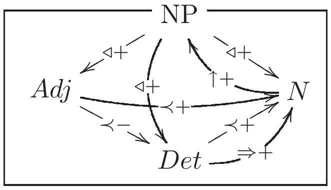


Fig. 1b

Figure 1b shows a more complete graph, corresponding to an entire sentence. Again, no relation in this graph plays a specific role. The information comes from the fact that this set of categories are linked by several relations. The set of relations forms a description: it tells us that linearity, requirement, obligation, constituency properties are satisfied, they characterize an *S.* Theoretically, each node can be connected to any other node. Nothing forbids for example to represent a relation of some semantic type between the adjective and the verb nodes. By another way, when taking from this graph constituency relations only, we obtain a dominance tree:



Finally, insofar as a property can be satisfied or violated in a characterization, we have to label relations with their type and their interpretation (true or false, represented + or -). The following example presents a graph for the assignment A=*{NP, Adj, Det, N}*, in which the determiner has been realized after the adjective.



In this graph, all constraints but the precedence between *Det* and *Adj* have been satisfied, the corresponding relations being labeled with +.

6 Explaining levels of grammaticality

As a side effect, representing information in this way also constitues a possibility to rank the inputs according to a grammaticalness evaluation. We present in this section how to use characterizations in order to quantify such information. The idea (see [Blache & al. 06]) consists in analyzing the form of the graph and its density, taking into account the interpretation of the relations. More precisely the method consists in calculating an index from the cardinality of P^+ and P^-, (respectively the set of satisfied and violated properties). Let's call N^+ and N^- the cardinality of these sets. The first indication that can be obtained is the ratio of satisfied properties with respect to the total number of evaluated properties E. This index is called the *Satisfaction ratio*, calculated as $SR = \frac{N^+}{E}$.

Going further, it is also possible to give an account of the coverage of the assignment by means of the ratio of evaluated properties with respect to the total number of properties T describing the category in the grammar. This coefficient is called *Completeness coefficient*: $CC = \frac{E}{T}$.

A *Precision Index* can to its turn be proposed, integrating these two previous information: $PI = \frac{SR+CC}{2}$.

Finally, a general index can be proposed, taking into consideration the different indexes of all the constituents. For example, a phrase containing only well-formed constituents has to be assigned a higher value than one containing ill-formed ones. This is done by means of the *Grammaticalness Index*, d being the number of embedded constructions C_i: if $d = 0$ then $GI = PI$, else $GI = PI \times \frac{\sum_{i=1}^{d} GI(C_i)}{d}$.

In reality, this different figures need to be balanced with other kind of information. For example, we can take into consideration the relative importance of constraint types in weighting them. Also, the influence of SR and CC over the global index can be modified by means of coefficients.

This possibility of giving a quantified estimation of grammaticalness directly comes from the possibility of representing syntactic information in a fully contraint-based manner, that has been made possible thanks to the MTS view of grammar.

7 Conclusion

The representation of syntactic information by means of constraints, as described in this paper, has several advantages. First, it makes it possible to implement the entire MTS programme in which derivation does not play anymore a role. The shift from generative to model-based conception of syntax becomes then concrete: constraint satisfaction completely replaces derivation. This evolution becomes possible provided that we abandon a strict hierarchical representation of syntax in which dominance plays a central role.

As a consequence, such fully constraint-based approach offers the possibility to replace ordered trees domain with that of constraint graphs. This is not only a matter of representation, but has deep consequences on theory itself: different types of information is represented by different relations, all of them being at the same level.

The *Property Grammar* framework described in this paper represents the possibility of an actual MTS implementation in which constraints are not only a control layer over the structure, but represent the structure itself: MTS is not GES plus constraints, provided that dominance is not represented separately from other information.

References

[Aarts07] Aarts. B (2007) *Syntactic Gradience. The nature of Grammatical Indeterminacy*, Oxford University Press.

[Blackburn & al. 93] Blackburn P., C. Gardent & W. Meyer-Viol (1993) “Talking About Trees” in proceedings of *EACL*

[Blackburn & Meyer-Viol 94] Blackburn P. & W. Meyer-Viol (1994) “Linguistics, Logic and Finite Trees” in *Bulletin of the IGPL*, 2.

[Backofen & al. 95] Backofen R., J. Rogers, and K. Vijay-Shanker (1995) “A First-Order Axiomatization of the Theory of Finite Trees” in *Journal of Logic, Language, and Information*, 4:1

[Blache05] Blache P. (2005) “Property Grammars: A Fully Constraint-Based Theory", in H. Christiansen & al. (eds), *Constraint Solving and NLP*, Lecture Notes in Computer Science, Springer.

[Blache & al. 06] Blache P., B. Hemforth & S. Rauzy (2006), “Acceptability Prediction by Means of Grammaticality Quantification”, in proceedings of *COLING-ACL 06*

[Chomsky75] Chomsky N.. (1975) *The Logical Structure of Linguistic Theory*, Plenum Press

[Cornell & Rogers 00] Cornell T. & J. Rogers (2000) “Model Theoretic Syntax”, in L. Lai-Shen Cheng & R. Sybesma (eds), *The Glot International State of the Article Book I*, Holland Academic Graphics

[Gazdar & al. 85] Gazdar G., E. Klein, G. Pullum & I. Sag (1985) *Generalized Phrase Structure Grammars*, Blackwell

[Huddleston & Pullum 02] Huddleston R. G. Pullum (2002) *The Cambridge Grammar of the English Language*, Cambridge University Press.

[Keller00] Keller F. (2000) *Gradience in Grammar. Experimental and Computational Aspects of Degrees of Grammaticality*, Phd Thesis, University of Edinburgh.

[Prince93] Prince A. & Smolensky P. (1993) *Optimality Theory: Constraint Interaction in Generative Grammars*, Technical Report RUCCS TR-2, Rutgers Center for Cognitive Science.

[Pullum & Scholz 01] Pullum G. & B. Scholz (2001) “On the distinction between model-theoretic and generative-enumerative syntactic frameworks”, in processdings of the conference on *Logical Aspects of Computational Linguistics*, Springer

[Pullum & Scholz 03] Pullum G. & B. Scholz (2003) “Foundations of Model-Theoretic Syntax”, Introductory course of the *ESSLLI-03*

[Rogers97] Rogers J. (1997) “Grammarless Phrase Structure Grammar”, in *Linguistics and Philosophy*, 20

[Sag al. 03] Sag I., T. Wasow & E. Bender (2003) *Syntactic Theory. A Formal Introduction*, CSLI.