Model-Theoretic Syntax: Property Grammar, Status and Directions

Philippe Blache, Jean-Philippe Prost

Constraints and Language

Philippe Blache Henning Christiansen
Verónica Dahl Denys Duchier
Jørgen Villadsen (Eds.)

April 6, 2014

CONTENTS

CHAPTER ONE

MODEL-THEORETIC SYNTAX: PROPERTY GRAMMAR, STATUS AND DIRECTIONS

PHILIPPE BLACHE & JEAN-PHILIPPE PROST

1.1 INTRODUCTION

The question of the logical modelling of natural language is concerned with providing a formal framework, which enables representing and reasoning about utterances in natural language. The body of work in this area is organised around two different hypotheses, which yield significantly different notions of what the object of study is. Each of those two hypotheses is based on a different side of Logic: the proof-theoretic hypothesis, and the model-theoretic one. The proof-theoretic hypothesis, on one hand, considers that natural language can be modelled as a formal language. It sets the syntax of the observed natural language at the syntax level of the modelling formal language. All the works based on Generative Grammar rely on this hypothesis. The model-theoretic hypothesis, on the other hand, considers that natural language, along all its dimensions including Syntax, must be modelled through the semantic level of Logic. Underlying are two fairly different notions of what natural language is — and is not, and what should — or not — be modelled.

We introduce and compare the main characteristics of both the Proof-Theoretic and the Model-Theoretic paradigms. We argue that representing the linguistic description of an utterance solely through a hierarchical syntactic structure is severely restrictive. We show evidence of these restrictions through the study of specific problems. We then argue that a Model-Theoretic representation of Syntax does not show those restrictions, and provides a more informative linguistic description than Proof-Theoretic Syntax. We give an overview of a specific framework for MTS, called *Property*

Grammar (PG), which we use to illustrate our point. We show, in particular, how to rely on a graph as linguistic representation, in order to address various language problems.

1.2 MODEL THEORY FOR MODELLING NATURAL LANGUAGE

The Proof-Theoretic and the Model-Theoretic approaches to modelling Natural Language differ in scope of modelling, i.e. with regard to the observations being modelled. They also differ with regard to the nature of the linguistic knowledge being captured and represented.

Proof-Theoretic Syntax With the proof-theoretic one, natural language is defined as the set $\mathscr{L}(G)$[1] of all the strings licensed by the grammar G. What is meant here by *licensed* is *proven*: all the strings in $\mathscr{L}(G)$ are those, and only those, which can be proven by a set of production rules from G. The proof itself captures the whole linguistic knowledge about any string s in $\mathscr{L}(G)$, in the form of a tree representation: the *parse tree* — or syntactic structure. Thus, a parse tree for the sentence s is merely a graphic (in the sense of the Graph Theory) representation of the proof that $s \in \mathscr{L}(G)$. This isomorphism between linguistic structure and proof of membership has strong consequences on the modelling scope, since anything that can not be proven can not be represented either. Therefore, and assuming that a grammar G is available, which captures all the observed canonical linguistic phenomena of human language, the set of all the objects being modelled under the proof-theoretic hypothesis is limited to the set of the grammatical strings, and only these ones. All ungrammatical strings are just out of scope, and no knowledge can be represented about them. This notion is very restrictive, for it does not account for the extreme variability of language usages, including non-canonical or even ill-formed productions.

Furthermore, the linguistic description being represented is heavily driven by Syntax, and does not (or rarely) account for other linguistic dimensions. Even when it does, the information on syntax is required in order for information on other dimensions to be represented. Recent works, on

[1] More formally, the language $\mathscr{L}(G)$ is usually defined as the n-uple $\langle L, C, S, G\rangle$, where L is a lexicon (terminal vocabulary), C is a set of morpho-syntactic categories (non-terminal vocabulary), $S \in C$ is a start symbol (for labelling the tree root), and G (the grammar) is a set of production rules.

the contrary, propose to consider all different sources of information as interacting at the same level (see, in particular, works around Construction Grammar, e.g. Goldberg (2003), Sag (2012)).

Model-Theoretic Syntax The model-theoretic notion of language, on the other hand, does not show those limitations in scope. Here, no assumption is made as to which utterances ought to be described — hence covered by the model, and which ones ought not to be. All observed utterances get a linguistic description of their structure, irrespective of their well-formedness. The description takes the form of a set of grammar statements, each of them being either verified or not by the structure. All those grammar statements together, instantiated for a given utterance, constitute a constraint network. The grammaticality of an utterance is then defined with respect to this description: the utterance is grammatical if and only if all the statements are verified. Meanwhile, descriptions which would include failing statements may still exist — they are simply not deemed grammatical. Hence scope-wise, any observed utterance may get a linguistic description.

Another incentive of a model-theoretic representation of grammar is that it offers the possibility to interpret in different ways the knowledge at stake. For instance, in this chapter, we present two different (though not opposed) perspectives on what the representation of a linguistic structure should be like: the *constructive* perspective, and the *descriptive* perspective.

The Constructive Perspective on MTS The constructive perspective considers that the constraint network instantiated for an utterance is self-sufficient for describing the linguistic knowledge about this utterance. The network comes in replacement of the conventional linguistic phrase and dependency structures. Yet, if required, those structures may be recovered by induction from the network. This perspective also makes use of the MT grammar for the parsing process, which, therefore, solely relies on the constraint grammar for building up the constraint network through an inference process. Section 1.3 elaborates further on various aspects of such a perspective on MTS.

The Descriptive Perspective on MTS The descriptive perspective sees the constraint network as a complement of the conventional parse structure (phrase or dependency). The parse structure merely serves the purpose of the *structure*, in the sense of the Model Theory in Logic. According to it, the objects of study are *models* of *theories* expressed in a formal metalangage, where a *theory* is a set of statements specified in a formal language,

and a *model* of a theory is a structure, which satisfies all the statements of that theory. Section 1.4 elaborates further on the Descriptive perspective on MTS.

Either way (constructive or descriptive), the MT representation, inclusive of the constraint network, is much more informative than the sole parse structure, and allows for more exact reasoning on the corresponding utterance. The next two sections introduce those two perspectives.

1.3 THE CONSTRUCTIVE PERSPECTIVE: A CONSTRAINT NETWORK FOR REPRESENTING AND PROCESSING THE LINGUISTIC STRUCTURE

The generative conception of grammar relies on the derivation process which, in turn, depends on a hierarchical representation of syntactic information. However, several works have shown the limits of such a representation. From a generative point-of-view, parsing an input corresponds to finding a set of derivation rules, which makes it possible to generate the surface realisation of this input. This conception of grammar relies then on a specific view of what language is: the set of surface forms that can be generated by the grammar. This conception is very restrictive for several reasons. One is the extreme variability of language usages, including non canonical or even ill-formed productions. Another is the fact that this view is purely syntactically driven: only syntax is taken into account here and when other sources of information such as prosody are considered (which is rarely the case), they are considered as "complementary" to syntax, giving syntax a preeminent position. Recent works propose on the contrary to consider all different sources of information as interacting at the same level (see in particular works around Construction Grammar, Goldberg (2003), Sag (2012)).

Even though many linguistic theories now challenge this way of considering the relationship between language and grammar, most of them remain more or less based on the generative framework, in the sense that what we can call the context-free backbone still occupies a central position. Model-Theoretic Syntax Pullum and Scholz (2003), Pullum (2007) proposes a paradigmatic shift, making it possible to escape from this framework.

This section presents the main characteristics of these different conceptions of syntax. It describes more precisely the specific problems coming from the hierarchical conception of syntax, showing how it can constitute

a severe limitation for linguistic description. We propose then an overview of a specific 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.

1.3.1 GENERATIVE-ENUMERATIVE VS. MODEL-THEORETIC SYNTAX

There are two different approaches in logic: one is purely *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 for natural language syntax. A first approach (the syntactic one in logic) consists in studying sentence well-formedness. The problem consists there in finding a structure adequate to the input. In this case, grammatical information is then represented by means of a set a rules, the syntactic structure representing the set of rules used during the parse. An alternative approach consists in studying directly the linguistic properties of the sentence, instead of building a structure. Pullum and Scholz (2001) call these approaches respectively *Generative Enumerative Syntax* (GES) and *Model-Theoretic Syntax* (MTS). The first corresponds to the generative theories, it has been extensively experimented. The latter still remains less studied and only a few works belong to this paradigm .

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.

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 Chomsky (1975), 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 Chomsky (1975) (without proposing a solution), which forbids a finer conception, capable of representing in particular a grammaticality scale (also called *gradience*, see Keller (2000) or Aarts (2007)).

This generative conception of syntax is characterized as being enumerative (see Pullum and Scholz (2001) 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 Prince and Smolensky (1993)).

Model Theoretic Syntax proposes an alternative view (Blackburn *et al.* (1993), Cornell and Rogers (2000), Pullum and Scholz (2001)). 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 $\mathscr{K}$ a set of categories, a description of a nominative noun comes to the formula:

$$\exists x[cat(x,N) \wedge gend(x,masc)] \tag{1.1}$$

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:

$$[cat(x,Det) \wedge cat(y,Det) \rightarrow x \approx y] \tag{1.2}$$

Parsing 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 in turn the complex predicates formed by the grammar 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*.

Various parsing strategies have been implemented in line with that approach. Balfourier *et al.* (2002) implements an incremental strategy, where the choice of a suitable assignment relies on a heuristic of shortest possible span. VanRullen (2005) implements a multi-graph representation of the constraint network, and a strategy, which allows different granularities of solution, from chunks to deep parses. Prost (2008) revisit the CKY parsing algorithm, based on dynamic programming techniques, in order to optimise the proportion of properties (instead of probabilities) violated by the solution parse. More recently, Duchier *et al.* (2010) explore the possibility to model the parsing problem according to a model-theoretic grammar as a Constraint Optimisation Problem.

Finding a model is, then, completely different from deriving a structure. As stressed by Pullum and Scholz (2001), instead of enumerating a set of expressions, an MTS grammar simply states conditions on these expressions.

Model-Theoretic Syntax (herafter MTS) moves then from a classical tree domain to 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, where all constraints are at the same level and evaluated independently from each others (no order is enforced on the constraints for evaluation).

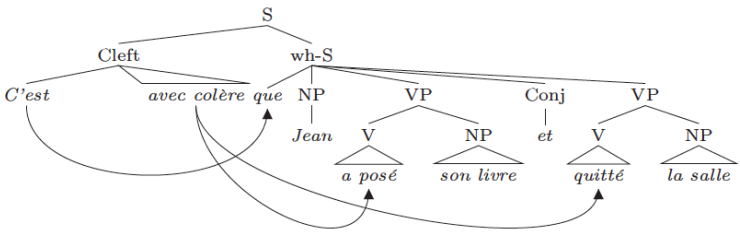
1.3.2 GENERATIVITY AND HIERARCHICAL STRUCTURES

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 *et al.* (2003)), 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.

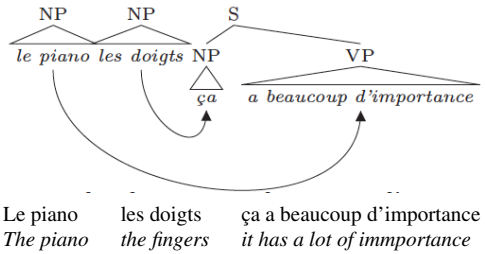
However, in many cases, a representation in terms of tree is not adapted or even not possible. The following example illustrate this situation. It present the case of a cleft element adjunct of two coordinated verbs.



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 exists 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:



This example illustrates 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 Keller (2000), Pullum and Scholz (2001)).

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.

1.3.3 THE PROPERTY GRAMMAR FRAMEWORK

A seminal idea in GPSG (see Gazdar *et al.* (1985)) 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 Blache (2005)) 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 *et al.* (1995) for more precise definitions). We use the set of binary predicates for immediate domination ($\lhd$), 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:

- $Const(A,B) : (\forall x,y)[(A(x) \wedge B(y) \rightarrow x \lhd y]$

  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.

- $Excl(A,B) : (\forall x)(\not\exists y)[A(x) \wedge B(y)]$

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 grammar 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 and Scholz (2003)).

Parsing with Property Grammar 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, in order to infer new categories and build up the parse structure. 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 as 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 inferred.

In terms of representation, unlike the classical approaches, syntactic information is not represented by means of a tree (see Huddleston and Pullum (2002)), 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), $\lhd$ (constituency), $\uparrow$ (obligation), $\prec$ (precedence). A *Property Grammar* graph (noted *PG-graph*) is a tuple of the form:

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

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 the use of such graphs in section 1.5.

1.4 THE DESCRIPTIVE PERSPECTIVE: A CONSTRAINT NETWORK FOR COMPLETING THE LINGUISTIC STRUCTURE

According to the model-theoretic hypothesis, human language is represented on the semantic level of Logic: the objects of study are *models* of *theories* expressed in a formal language, where a *theory* is a set of statements specified in a formal meta-language, and a *model* of a theory is a structure, which satisfies all the statements of that theory. Hence, applied to natural language:

- a theory is a set of grammar statements, specified by a conjunction $\Phi = \bigwedge_i \phi_i$, where every atom ϕ_i is a logical formula, which puts elements of the structure in a relationship ;
- a structure is a linguistic parse structure (e.g. phrase structure, dependency structure, or both).

A grammar is, then, a conjunctive formula, parameterised by the structure, and a theory is an instance of the grammar for a given structure. For instance, for a domain of phrase structures, the ϕ_i are relations, which hold on constituents (e.g. *In a Noun Phrase in English, the Determiner precedes the Noun*). Duchier *et al.* (2009) formulate a Model-Theoretic semantics for Property Grammar along these lines.

Model checking We first give a few definitions, in order to fix the notations in use in the following.

Let $\mathscr{S}$ be a set of words in the target language, and $\mathscr{L}$ a set of labels, which denote morpho-syntactic categories; a *lexicon* is then a subset $V \subset \mathscr{L} \times \mathscr{S}$ (which implicitly assumes that the terminals are POS-tagged words). Let $\mathscr{P}_{\mathscr{L}}$ be the set of all the possible properties on $\mathscr{L}$; a PG grammar Φ is specified by a pair (P_G, L_G), with $P_G \subseteq \mathscr{P}_{\mathscr{L}}$.

Let $\tau : s$ be a (phrase structure) tree decorated with labels in $\mathscr{L}$, and whose surface realisation is the string of words s; let Φ^s be an instantiation of Φ for $\tau : s$; $\tau : s$ is a model for Φ^s iff $\tau : s$ makes Φ^s true. We denote by $\tau : s \models \Phi^s$ the satisfaction of Φ^s by $\tau : s$. The instantiation Φ^s is also called the *constraint network* associated with $\tau : s$ for the grammar Φ.

Definition 1 (*Grammaticality*)**.** $\tau : s$ is grammatical with respect to the grammar Φ iff $\tau : s \models \Phi^s$

Since $\Phi^s = \bigwedge_i \phi_i^s$, Definition 1 means that every instance of property ϕ_i^s of Φ^s for the sentence s must be satisfied for s to be deemed grammatical with respect to the grammar Φ.

The model checking process involves:

- instantiating the grammar Φ for the parse tree $\tau : s$,
- building up the corresponding constraint network Φ^s, and
- checking the truth of every atom ϕ_i^s.

Processing-wise, the existence of a linguistic structure is required prior to checking it against a grammar — which involves constructing the constraint network. Figure 1.4 exemplifies a phrase structure deemed ungrammatical through model checking.

Under such an interpretation, the linguistic structure plays the role of a semantic object, which makes a *theorie* true (in which case the structure is deemed a *model* of the theory), or not. That is, the parse structure is the object, which makes the grammar (as a conjunction of statements) true, or not. Here, the conventional parsing process, seen as the generation process of the parse structure, is kept separate from the model checking process.

Language cover As far as knowledge representation is concerned, a formal framework for MTS shows the following properties:

1. the independence of the grammar statements allows for the definition of non-classical satisfaction, whereby a given structure only partially satisfies a subset of Φ, and violates its complement;

```
419:  En_effet, sept projets sur quatorze, soit la moitié, ont
      un financement qui n' est toujours pas assuré et dont le
      calendrier n' est pas_encore arrêté.
( (SENT (ADV En_effet) (PUNC ,)
        (NP (DET sept)
            (NC projets)
            (PP (P sur)
                (NP (NC quatorze)))
            (PUNC ,)
            (COORD (CC soit)
                (NP (DET la) (NC moitié)))) (PUNC ,)
        (VN (V ont))
        (NP (DET un)
            (NC financement)
            (Srel
                (NP (PROREL qui))
                (VN (ADV n')
                    (V est)
                    (AdP (ADV toujours) (ADV pas))
                    (VPP assuré))))
        (COORD (CC et)
            (Sint
                (NP (NC dont) (DET le) (NC calendrier))
                (VN (ADV n') (V est) (ADV pas_encore) (VPP arrêté))))
    (PUNC .)))
```

Figure 1.1: Example of parse deemed ungrammatical through model checking

2. *how* structures are generated is not specified in the grammar.

Property 1 means that *any* structure may be represented in the framework, whether grammatically satisfactory or not. The syntax of non-canonical language can, thus, be modelled by a structure, which only loosely meets the grammar. Property 2 means that processing-wise, the generation of candidate-model structures is formally distinct from the grammar check. It opens all kinds of perspectives with regard to the parsing process. We have already seen in section 1.3 that the entire parsing process, including the generation of the structure, may be driven by the PG grammar itself. Another option, in line with the model theory, is to consider that the generation of the candidate structures is not of concern to the theory. In this case, different options are available.

Generation of model-candidates Since a model-theoretic representation is independent from any processing aspects regarding the generation of candidate structures, the strategy for generating them may be conceived separately from model checking. Although an inference process may be designed in order to construct structures on the sole basis of the constraint grammar (see the Constructive perspective 1.3, and for instance Maruyama (1990); Balfourier *et al.* (2002); Prost (2008); Duchier *et al.* (2010) for parsing strategies), nothing makes it compulsory. It is, therefore, possible, for instance, to check likely structures generated by a probabilistic parser against the MT grammar.

Note, as well, that the type of linguistic structure concerned by a Model-Theoretic representation may take different forms, depending on the formal framework in use. The seminal work of Maruyama (1990), for instance, is concerned with dependency structure, while Optimality Theory Prince and Smolensky (1993) is more used for describing phonological structures. As for Property Grammar, the framework is essentially used with phrase structures, though a few works rely on it for multimodal annotation, or biological sequences analysis.

Structure enrichment for solving language problems A major incentive of the descriptive perspective on MTS, and PG in particular, stands in its potential alliance with other structures and processes in order to address a variety of language problems. Grammaticality judgement is one of those. The problem is encountered in contexts such as Statistical Machine Translation Zwarts and Dras (2008), Summarisation Wan *et al.* (2005), or second language learning Wong and Dras (2011), where the grammaticality of a candidate solution is a non-trivial decision problem. A common approach to address it is to train a statistical classifier Foster *et al.* (2008); Wong and Dras (2011); Wagner (2012) in order to determine the grammaticality of a candidate around a threshold of likelyhood. The use of a constraint network, associated with the linguistic structure, alleviates the decision through exact model checking (Prost, forthcoming).

The graph-based representation also shares properties with the graph structure of semantic networks, such as Lafourcade's 2008. The combination of the two is expected to open avenues of research with respect to the Syntax-Semantics interface. The solving of problems concerned with both dimensions, such as Grammar error detection, should be eased by the resulting enriched structure Prost and Lafourcade (2001).

1.5 GRAMMATICALITY JUDGMENT

Classically, syntactic information is usually represented in terms of decorated ordered trees (see Blackburn *et al.* (1993), Blackburn and Meyer-Viol (1994)). 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.

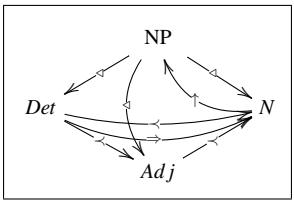


Fig. 1a

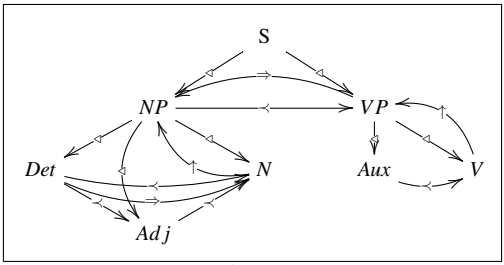
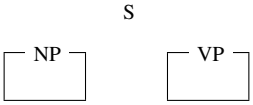
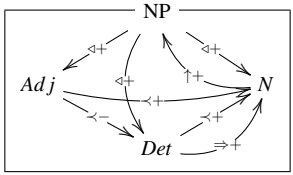


Fig. 1b

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 +.

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 *et al.* (2006)) 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.

1.6 CONCLUSION

The representation of syntactic information by means of constraints, as described in this paper, shows several advantages. First, it provides an elegant computational framework for MTS, where derivation does not play any role. In the Constructive perspective on MTS, the shift from generative to model-based conception of natural language syntax then becomes con-

crete: 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 Descriptive perspective on MTS, where the constraint networks complements and enriches the conventional lingustic struture, also shows interesting properties. Through the provision of a finer-grained description of the linguistic properties of a sentence than the sole parse structure, it alleviates the address of various language problems. Grammaticality judgement, for instance, or the interaction between the Syntax and Semantics dimensions, should benefit from such a graph-based representation.

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.

BIBLIOGRAPHY

Aarts, B. (2007). *Syntactic gradience: the nature of grammatical indeterminacy*. Oxford University Press.

Backofen, R., Rogers, J., and Vijay-Shanker, K. (1995). A first-order axiomatization of the theory of finite trees. *Journal of Logic, Language, and Information*, **4**(1).

Balfourier, J.-M., Blache, P., and Rullen, T. V. (2002). From Shallow to Deep Parsing Using Constraint Satisfaction. In *Proc. of the 6th Int'l Conference on Computational Linguistics (COLING 2002)*.

Blache, P. (2005). Property grammars: A fully constraint-based theory. In H. C. et al., editor, *Constraint Solving and Language Processing*, volume LNAI 3438. Springer.

Blache, P., Hemforth, B., and Rauzy, S. (2006). Acceptability prediction by means of grammaticality quantification. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual*

Meeting of the Association for Computational Linguistics, pages 57–64, Sydney, Australia. Association for Computational Linguistics.

Blackburn, P. and Meyer-Viol, W. (1994). Linguistics, logic and finite trees. *Bulletin of the IGPL*, **2**.

Blackburn, P., Gardent, C., and Meyer-Viol, W. (1993). Talking about trees. In *Proeedings of EACL-93*.

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

Cornell, T. and Rogers, J. (2000). Model theoretic syntax. In C. L. Lai-Shen and R. Sybesma, editors, *The Glot International State of the Article Book I*. Holland Academic Graphics.

Duchier, D., Prost, J.-P., and Dao, T.-B.-H. (2009). A model-theoretic framework for grammaticality judgements. In *Formal Grammar*, pages 17–30.

Duchier, D., Dao, T.-B.-H., Parmentier, Y., and Lesaint, W. (2010). Property grammar parsing seen as a constraint optimization problem. In *FG*, pages 82–96.

Foster, J., Wagner, J., and van Genabith, J. (2008). Adapting a wsj-trained parser to grammatically noisy text. In *ACL (Short Papers)*, pages 221–224.

Gazdar, G., Klein, E., Pullum, G., and Sag, I. (1985). *The Logic of Typed Feature Structures*. Blackwell.

Goldberg, A. (2003). Constructions: A new theroretical approach to language. *Trens in Cognitive Sciences*, **7**(5), 219–224.

Huddleston, R. and Pullum, G. K. (2002). *The Cambridge Grammar of the English Language*. Cambridge University Press.

Keller, F. (2000). *Gradience in Grammar - Experimental and Computational Aspects of Degrees of Grammaticality*. Ph.D. thesis, University of Edinburgh.

Lafourcade, M. and Joubert, A. (2008). Evolutionary Basic Notions for a Thematic Representation of General Knowledge. In *proceedings of LREC'08: Language Resources and Evaluation Conference*.

Maruyama, H. (1990). Structural Disambiguation with Constraint Propagation. In *Proceedings 28th Annual Meeting of the ACL*, pages 31–38, Pittburgh, PA.

Prince, A. and Smolensky, P. (1993). Optimality Theory: Constraint Interaction in Generatire Grammar. Technical report, TR-2, Rutgers University Cognitive Science Center, New Brunswick, NJ.

Prost, J.-P. (2008). *Modelling Syntactic Gradience with Loose Constraint-based Parsing*. Ph.D. thesis, Macquarie University, Sydney, Australia, and Université de Provence, Aix-en-Provence, France (cotutelle).

Prost, J.-P. and Lafourcade, M. (2001). Pairing Model-Theoretic Syntax and Semantic Network for Writing Assistance. In *Proceedings of CSLP 2011 (Constraints and Language Processing)*, pages 56–68, Karlsruhe, Germany.

Pullum, G. (2007). The evolution of model-theoretic frameworks in linguistics. In J. Rogers and S. Kepser, editors, *Proceedings of Model-Theoretic Syntax at 10 Workshop*.

Pullum, G. and Scholz, B. (2001). On the Distinction Between Model-Theoretic and Generative-Enumerative Syntactic Frameworks. In P. de Groote, G. Morrill, and C. Rétoré, editors, *Logical Aspects of Computational Linguistics: 4th International Conference*, number 2099 in Lecture Notes in Artificial Intelligence, pages 17–43, Berlin. Springer Verlag.

Pullum, G. K. and Scholz, B. C. (2003). Model-Theoretic Syntax Foundations - Linguistic Aspects. Draft; ask for authors' written consent prior to citation or quotation.

Sag, I. (2012). *Sign-Based Construction Grammar: An Informal Synopsis*, pages 39–170. CSLI.

Sag, I., Wasow, T., and Bender, E. (2003). *Syntactic Theory. A Formal Introduction*. CSLI.

VanRullen, T. (2005). *Vers une analyse syntaxique à granularité variable*. Ph.D. thesis, Université de Provence, Informatique.

Wagner, J. (2012). *Detecting Grammatical Errors with Treebank-Induced, Probabilistic Parsers*. Ph.D. thesis, Dublin City University, Dublin, Ireland.

Wan, S., Dras, M., Dale, R., and Paris, C. (2005). Towards Statistical Paraphrase Generation: Preliminary Evaluations of Grammaticality. In *Proceedings of The 3rd International Workshop on Paraphrasing (IWP2005)*.

Wong, S.-M. J. and Dras, M. (2011). Exploiting Parse Structures for Native Language Identification. In *EMNLP*, pages 1600–1610.

Zwarts, S. and Dras, M. (2008). Choosing the right translation: A syntactically informed classification approach. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 1153–1160. Association for Computational Linguistics.