Properties in 5P

Gabriel G. Bès, Caroline Hagège

Properties in 5P

Gabriel G. Bès [1], Caroline Hagège [2]
[1] GRIL, Université Blaise-Pascal, Clermont-Ferrand
[2] Xerox Research Centre Europe, Meylan

Research Report, GRIL, Université Blaise-Pascal, Clermont-Ferrand, November 22, 2001

Abstract

This research report presents a formalism for the description of natural languages, called *Properties*. Properties are an alternative to formal grammars such as GPSG, HPSG, LFG, etc. A grammar of this type has a twofold function: it is intended to be both a description of some language and the declarative source of some algorithmic machinery that can process that language. Properties, on the contrary, allow to distinguish the two aspects and are strictly limited to the descriptive aspect. They consist in formulas about the categories in the strings of the described language, specifying uniqueness and optionality characteristics, and cooccurrence, agreement or linearity relations.

Properties, as a formalism, are part of a larger framework: the 5P paradigm. The 5P paradigm is conceived as a mean to effectively place linguistics within the class of empirical sciences. It is outlined in section 2 and further discussed in section 9.

Properties and the 5P paradigm were first defined in (Bès, 1999) – see the references below. They are a development of concepts defined earlier in (Bès and Jurie, 1992).

See also

Gabriel G. Bès, « La phrase verbale noyau en français », *Recherches sur le français parlé*, vol. 15, Publications de l'Université de Provence, p. 237-358, 1999. <hal-01005527>

Pierre-François Jurie, Gabriel G. Bès, "The Control of UCG Grammars", *in* Gabriel G. Bès, Thierry Guillotin (eds), *A Natural Language and Graphics Interface: Results and Perspectives from the ACORD Project*, Springer, p. 47-64, 1992. <hal-01101333>

Properties in 5P

Gabriel G. Bès[*]
Universitè Blaise-Pascal - GRIL

Caroline Hagège[†]
Xerox Research Centre Europe

November 22 2001

1 Introduction

The goals of the 5P Paradigm, at least in a general formulation, are much the same than the ones, explicit or not, of other linguistic approaches to the study and/or processing of natural languages. The goals are :

- to describe natural languages;
- to explain natural languages;
- to process natural language utterances either in the task of acquisition of a natural language or in the use of a natural language, and, in this case, either in the interpretation of utterances -i.e. analysis- or in the production of utterances -i.e. generation;
- to develop a principled technology on natural languages; i.e. a technology explicitely related to the previous three points.

In this document we present an abstract of 5P in order to give a general view of its different components and, with more detail, of one of them, namely Properties (P2, see below)[1].

If 5P is labelled *a paradigm*, and not simply *a theory* or *a model* in more current terminology, it is because the general goals above are tackled in 5P in particular ways and with particular

[*] Gabriel.Bes@univ-bpclermont.fr; 34 Ave. Carnot, F 63037 Clermont-Fd Cedex.

[†] Caroline.Hagege@xrce.xerox.com; 6 chemin de Maupertuis, F 38240 Meylan.

[1] In documents published in 1999 and listed in Section 9.7 below, it was mentioned an internal and incomplete GRIL report draft with the title "The 5P Paradigm", cosigned with Philippe Blache. It was intended to become published within the general framework of a collaborative project that we three (i.e. the two authors of the present document and him) formally sketched in the 98 summer. Our colleague, at the end of 1999, in order to obtain an academic degree, presented to Paris VII University his folder *Contraintes et théories linguistiques: des Grammaires d'Unification aux Grammaires de Propriétés*. In this document he incorporates some aspects of the 5P Paradigm, while diverging significantly in others, as it was pointed out in [Bès 99f]. Blache's work is now published in Philippe Blache, *Grammaire de Propriétés* (Paris, Hermes, 2001), which has not yet been thoroughly examined. More generally, in the present document, we do not detail both what our view owes to other linguistic insights and in what it differs from them. We develop in it (Sections 6 to 9 below) some aspects already sketched in the "The 5P Paradigm" draft of 1999 with a more formal presentation of the Proprieties, the definitions of which were already given in a much less formalised version in [Bès 99a]. Futhermore, we add a succinct general overwiew of 5P in Sections 1 to 5, and, in Section 9 a presentation of the antecedents of 5P and of work in the 5P pattern, published or not, including ungoing one and the blibliographic references of this document, with possible access to some of them via Internet.

requirements, which are only sketched here and will be presented in a more detailed way in another document. We emphasize here that the underlying basic ambition of 5P is to effectively range linguistics within the class of empirical sciences, understanding by *empirical* not some inductive method to construct hypothesis from sensation, but rather a characterisation of the class of sciences which includes physics, biology and chemistry as classic illustrations, and excludes mathematics and logics, which are ranged in the so called formal sciences.

5P is not a closed and well defined package. It is not a well defined package at the time being and it is not intended to become one some day. Rather it is a frame enabling to integrate different and today dispersed insights about natural languages, and, when coherent integration is not possible, enabling to compare them, inasmuch as problems and offered solutions are presented in non-dogmatic terms.

The below 5P presentation must thus be understood in a dynamic perspective. We think that the work that has already been done within the framework justifies this presentation, but much work has to be done yet in order to be able to evaluate it accurately, and in order to refine, extend or complete it.

2 A general outline of 5P

Each one of the five P's defines a kind of module of the Paradigm : *P1* or *P* of Protocoles; *P2* or *P* of Properties; *P3* or *P* of Projections; *P4* or *P* of Principles; *P5* or *P* of Processes.

- *Protocoles* are observations sytematically captured and represented in some language.
- *Properties* are purely declarative descriptions of strings of some kind of units of a particular natural language.
- *Projections* are generalisations over Properties or subsets of Properties of some natural language, or of a reduced set of natural languages, or over strings defined by Properties.
- *Principles* are cross linguistic constraints on Projections and/or Properties of a significant class of natural languages.
- *Processes* are effective computational procedures on strings of natural languages.

In the classic inductive-deductive pattern of empirical sciences there is some constant interaction between observation, formulation of hypothesis, making deductions from hypotheses, testing of deductions with respect to observations, these ones either already registered before the hypotheses have been formulated or to be captured and registered after.

It is possible to situate with respect to the inductive-deductive pattern of empirical sciences the P's of the 5P Paradigm. Protocoles are registered observations. Properties are formal descriptions, much like axioms, from which, formally, deductions can be obtained and tested with respect to Protocoles. Projections and Principles are higher level descriptive statements than Properties - sometimes there are said to be *explanations* - from which Properties can be deduced; in the best cases the yet to be spelled Principles could be considered general laws on natural languages.

Processes are, on one hand, effective computational tools which must effectively calculate deductions from descriptions, and, particularly, be able to integrate in their effective calculus, descriptive statements of different levels of generality (i.e., to effectively calculate from Properties, Projections and Principles). But Processes are also, on the other hand, the necessary link between descriptions and any kind of technology on natural languages.

3 Properties

Suppose we have the following strings and the information labelled *1* or *0* associated to each of them as indicated in the following.

(1)

b, c	1
c, c	1
c, g	1
b, g	0

It is possible to describe the strings in (1) saying the following with respect to all strings associated with *1*:

(2)

i there is an element from V = {b, c, g};
ii there is a c;
iii there is only one b and only one g;
iv b precedes c;
v c precedes g.

Given V^+ - i.e. the set of not null strings which can be formed from the set *V* - , the description in (2) is explicit enough if we want to discriminate among the strings of V^+ those which must be associated to *1* and those which must not. For example the strings *[b]* , *[g]* and *[b, g]* must not be associated to *1* because they do not satisfy (ii), and the string *[c, b]* because it does not satisfy (iv).

The description in (2) is a possible way of describing strings of a language. It is not the only way and it is not the most common way of doing so. In 5P we want descriptions of languages as illustrated in (2), that is descriptions with no algorithmic side effects. In (2) there is no instruction to concatenate. We have there, in (i) a Vocabulary *V* with elementary expressions, and in (ii) to (iv) what we called *Properties*, i.e. statements on strings of elements from *V*. There is nothing which compels us to evaluate the Properties spelled in (ii) to (iv) in one order or another. We contend that (2) is a purely declarative way of stating the description of a language.

A grammar, as the concept is used in GPSG, HPSG, LFG, categorial grammars, etc. is an object with a twofold function : it is intended to be both a description of some language and the declarative source of some algorithmic machinery that, given some string of the language, must analize it in order to obtain some kind of representation, and/or, given some kind of input, must generate the corresponding string(s) of the language. The syntax for expressing the rules of a grammar is thus the same as that used for expressing the declarative source that the algorithm must evaluate in order to analyze and/or to generate strings of symbols.

Within this restricted and widely used concept of grammar, in 5P there is no grammar at all. Properties, though being calculable, are not intended to be used as the declarative source of some effective and efficient parsing or generating algorithm. Algorithms belong to Processes (P5); their declarative sources must be formally derivated (or at least in principle derivable) from Properties, but the syntax to express the declarative sources is not necessarily the same syntax than the one used to express Properties.

The three following points sums up a general overall characterisation of descriptions in terms of P2.

- Properties are maximally factorised.
- A linguistic description in terms of P2 is intended to be an actual declarative perspective : no procedural aspects are dependent on the description itself.
- Given a particular string and a set of Properties, the target of linguistic analysis can be, and in interesting cases is, to point out, for each Property, if it is (i) satisfied by the input, (ii) not satisfied by the input, (iii) not relevant to the input.

The previous points can be illustrated by (2). As for the first two: in (2) the elements which can conform a well-formed string are indicated independently (i, ii, iii) from their linear relations (iv, v). There is nothing in (2) which compells to verify, given some string, (ii) before (iii) or the other way round. There is no instruction to concatenate at all.

As for the third characterizising point. Suppose we have the two following strings.

(3)

i [b, g]

ii [b, e, c, g]

We expect, given (2), that we can obtain a parse saying that (3.i) - i.e. the string (i) in (3) - satisfies Properties (i) and (iii), that it is not concerned by (iv) nor by (v) and that it does not satisfy (ii). Similarly, with respect to string (ii), we want to be said that Properties (ii) to (iv) are satisfied and that (i), as far as *e* is concerned, it is not.

String (3.ii) is, furthermore, an illustration of the issue of fluidity of observations. Suppose that depending on the domain, the social situation in which the language is used or any other factor, sometimes we want to describe the language with a possible *e* and sometimes we don't want to do so, and suppose that when *e* is in a string, it goes before *c*, but neither is it ordered with respect to *b* nor is it unique. In order to obtain a P2' set, different from (2) but related to it we must do the following:

- add the element *e* to *V*;
- add the precedence relation *e precedes c*.

Fluidity of observations is a typical situation in natural language descriptions. The underlying and basic idea is that it is possible to relate sets of P2 Properties by specifiyng membership on sets: it seems that, in many cases, it suffices to incorporate some element(s) to a given set or to withdraw some element(s) from a set, and, with no modification of the other memberships, it is possible to obtain descriptions which account for fluidity of observations.

Let us introduce the symbol *IX* as the identifier of the set Properties presented in (2). The Properties in (2) can thus be referred to as *Pr-IX*. The string *[b, c, g]* satisfies all the Properties in Pr-IX. We will say that *[b, c, g] is a model of Pr-IX* or, in abbreviated form, that *[b, c, g] is a m-IX*. Furthermore we say that *M-IX* is the set of all and only the models satisfying Pr-IX. In general, *a model* is a string of symbols that satisfies a finite set of Properties.

By the following we introduce the basic entities which we are working on and their notation.

- *cat* : variable on a set of feature/value pairs or, in simplified notation, of feature values;
- *IDn* : variable on a subclass of Property identifiers (Identifiers of Nuclear Properties[2]);
- *ID* : variable on a subclass of Property identifiers (Identifiers of non nuclear Properties);
- *Sm* : metavariable on the variables *ID* or *IDn*;
- *Sy* : metavariable on the variables *Sm*, *ID*, *IDn* or *cat*.

Example. Suppose that we have the values *n[oun], art[icle], adj1[adjective of type 1], adj2[adjective of type 2], card[inal], adv[erb]*, the French IDn's *Nn* (Nominal nuclear phrase), *ADJ1n* (Adjective nuclear phrase of type 1), *ADJ2n* (Adjective nuclear phrase of type 2), the French ID *N* (Nominal phrase) and the following strings in (4):

(4)

i trois fleurs;

ii très jolies;

iii très bleues;

iv trois très jolies fleurs;

v trois très jolies fleurs très bleues.

These strings can be respectively associated to the following models in (5).

(5)

i (card$_1$ n$_2$)$_{Nn}$

ii (adv$_1$ adj1$_2$)$_{ADJ1n}$

iii (adv$_1$ adj2$_2$)$_{ADJ2n}$

iv (card$_1$ ADJ1n$_2$ n$_3$)$_{Nn}$

v (Nn$_1$ ADJ2n$_2$)$_N$

[2] Given some phrase, Nuclear models, specified by Nuclear Properties, stand for strings from the initial element of the string to what is often called the head of the phrase. Thus, given the nominal phrase *the boy in the park*, *the boy* is associated to an *IDn* (*Nn*), and the whole phrase to an *ID* (*N*).

The above representations in (5) tells us simply which are the symbols used in each model and in what order, i.e. each symbol is related to a position in the model. If we want to express relations between symbols used in different positions in the model string, we can use arrows to express them. So we can obtain (6) or (6b), which are equivalent notations[3].

(6)

i (card$_{1\to 2}$ n$_{2\to 2}$)$_{Nn}$

ii (adv$_{1\to 2}$ adj1$_{2\to 2}$)$_{ADJ1n}$

iii (adv$_{1\to 2}$ adj2$_{2\to 2}$)$_{ADJ2n}$

iv (card$_{1\to 3}$ ADJ1n$_{2\to 3}$ n$_{3\to 3}$)$_{Nn}$

v (Nn$_{1\to 1}$ ADJ2n$_{2\to 1}$)$_{N}$

(6b)

i <(card$_1$ n$_2$)$_{Nn}$, {<1,2>, <2,2>}>

ii <(adv$_1$ adj1$_2$)$_{ADJ1n}$, {<1,2>, <2,2>}>

iii <(adv$_1$ adj2$_2$)$_{ADJ2n}$, {<1,2>, <2,2>}>

iv <(card$_1$ ADJ1n$_2$ n$_3$)$_{Nn}$, {<1,2>, <2,3>, <3,3>}>

v <(Nn$_1$ ADJ2n$_2$)$_{N}$, {<1,1>, <2,1>}>

In models of type (5) or of type (6) or (6b) we have the *model string*

(... Sy$_x$ ...)$_{Sm}$

where '...' stands for a variable on possibly null strings of *Sy's*, and the index *x* stands for a position in the model strings of models of the (5) type (i.e. *Basic models*, see below Section 5) or of the (6b) type (i.e. *Arrowed models*, see below), while, in Arrowed models as in (6), the index stands for a position arrowing to some position.

We can now sum up the three basic kinds of Properties that models must satisfy and what kind of information each one is intended to express.

- *Existence Properties*. They specify the sets of symbols which can be used in the different positions of a model string; for example we don't know of any set *{art, adv}* included in some set of symbols from which a model string not obtained by the application of the substitution rule (see below) can be specified in French, while, on the other hand, *{art, adv}* is indeed a set whose members can be used in model strings (e.g. *les trois*) obtained without the application of the substitution rule.
- *Linearity Properties*. They specify relations of order associated to a set of symbols specified by Existence Properties; e.g. French Linearity Properties will specify that in models in *M-Nn*, *art* precedes *card*.

[3] In each model with arrowing pairs we suppose an unique symbol arrowing to itself; see Section 8.

- *Arrowing Properties*. They specify sets of arrowing pairs, these being in the input to the *Semantic functions* intended to calculate semantic representations. Thus, the set of Arrowing pairs specify a graph, which is used by some function *F* of the set of Semantic functions. E.g.: in (6.i) Arrowing Properties specify the set *{<1, 2>, <2, 2>}* that is intended to be interpreted as saying : the compositional semantics associated to (6.1) must be calculated from the semantics associated to the symbols in position *1* and in position *2*, combined as stated by some *f* in the set of Semantic functions.

We said before that Properties are formal descriptions, much like axioms : this is why objects satisfying Properties are named *models*[4]. We assume that there are basically two kinds of axiomatic systems : with and without substitution rules. Properties are the analogs of axiomatic systems with substitution rules[5].

In general, a Substitution rule in 5P picks one symbol in a model string and changes it by some string of one or more symbols. We distinguish the *Model substitution rule* which substitutes for some symbol *Sm'* in the model string of some *m-Sm* a model string of some *m-Sm'*, and *Lexicon substitution rules*, which substitutes for some symbol *cat* in *m-Sm* an entry from the Lexicon (see next Section).

The use or not of Substitution rules allows to sharply characterize several subsets of *M-Sm*, among which the following two:

- models obtained without Substitution rules, which we note *M'-Sm;*
- models to which the Model substitution rule cannot apply because they do not have any *Sm* in their model string, which we note *pTM-Sm* (pre-Terminal Models).

Lexicon substitution rules apply to *pTM*: see below Section 5 *in fine*. Properties in this document are defined with respect to *M'-Sm.*

Lexicon substitution rules are formally simple: they put some symbol in the place of some other symbol of the input model, leaving all other elements in the input model unchanged. The Model substitution rule is more complex because positions in the model string are modified and, consequently, the notation of arrowing pairs is also modified. We do not formally present in this document the Model substitution rule. Rather, with the following examples, coming from the previous (4) to (6b), we give an intuitive idea of its function and use.

The models (5.v), (6.v), (6b.v) can be associated to string (4.v). The Model substitution rule is the mechanism for doing this. E.g. in Basic models (i.e. models without arrowing pairs), it substitutes (5.iv) for the symbol Nn_1 in (5.v), (5.ii) for the symbol $ADJ1n_2$ in (5.iv), and (5.iii) for the symbol $ADJ2n_2$ in (5.v), the result being:

$(\text{card}_1\ \text{adv}_2\ \text{adj1}_3\ \text{n}_4\ \text{adv}_5\ \text{adj2}_6)_N$

Observe that by the same token, the model associated to the following string (7.i) is (7.ii) after application of the Substitution rule which substitutes (5.i) for Nn_1 in (5.v).

[4] This concept was used from the beginning of the work with Pierre-François Jurie, see Section 9.1.

[5] In B.H.Partee, A.ter Meulen & R.E.Wall, *Mathematical Methods in Linguistics* (Dordrecht..., Kluwer, 1990, Section 8.2.1) the distinction is made between *axiomatic systems* and *extended axiomatic systems*. The expression of Proprieties, that requires substitution rules, belongs to the domain of *extended axiomatic systems*.

(7)

i trois fleurs très bleues

ii (card$_1$ n$_2$ adv$_3$ adj2$_4$)$_N$

Mutatis mutandis, we obtain the same effects by application of the Model substitution rule to arrowed models. Thus, given (7.i), we obtain its associated model, which follows, from the substitution of *Nn*$_1$ and *ADJ2n*$_2$ in (6b.v) by, respectively, (6b.i) and (6b.iii).

<(card$_1$ n$_2$ adv$_3$ adj2$_4$)$_N$, {<1, 2>, <2, 2>, <3, 4>, <4, 2>}>

After the presentation of the Lexicon in the next Section, we sum up in Section 5 the different types of models we can obtain. In Sections 6 to 8, the formalism of Properties related to *M'-Sm* is presented in some detail.

4 The lexicon

In (2.i) we introduced the notion of *V*, i.e. the set of elements from which strings are built. The analogous of *V* in 5P is a *Lexicon* (from hereafter *LE*), which is in fact a more sophisticated *V*.

Three sets - *LF, pRSEM, C* - are used in order to caracterize succintly *LE*.

LF is the set of linguistic forms. Each *linguistic form* (from hereafter *lf*) is just a string of characters. *LF* is a set of elementary objects - i.e. objects which are not obtained by any relation or rule whatsoever - and which are identified solely in terms of their graphic characteristics when printed[6]. An *lf* is thus a 'signifiant' in a Saussurian terminology.

pRSEM is the set of partial represented semantics, each *partial represented semantics* (from hereafter *prsem*) being the semantic representation assigned to some *lf*.

C is specified by system CAT.

A system CAT is the triplet

$$< FS, IR, top >$$

where *FS* stands for Feature set, *IR* for set of Inheritance relations (*Ir*) and *top* is a distinguished given category[7]. From *IR*, system CAT specifies an *IRG* (Inheritance relation graph) which in turn specifies the sets *C* and *MC*, such that $MC \subset C$, each element in *C* being a category and MC, a set of maximum categories (see immediately below); *mc* is the notation for a category in *MC*, and *cat* the notation for a category in *C*. That is, from <*FS, IR, top*>, *IRG* can be constructed and *C* and *MC* specified.

FS is a set of features. A feature is

$$< label; v_1, \ldots, v_n > or < label; VS >$$

where $v_1, \ldots, v_n = VS$.

A feature is thus a label associated with a set of values.

[6] In this presentation of *LF* we bypass all issues related to any kind of morphology. In a better finished one it is better to use lemmas as intermediates between effective occurrences and *mc's* (see in the text below) with or without *prsem's*. In this case we do not have in *LF* forms such as French *aimons, aimerait, aimé* but rather the lemma *aimer*.

[7] System Cat is specified in [Bès 01a].

Given two features in *FS*, $< label_i; VS_i >$ and $< labelj; VS_j >$, there is no *w* such that *w* $\in (VS_i \cap VS_j)$: a value *w* cannot be a value of two different labels. A label/value pair is noted *<label = value>*.

A category *cat* is a set of label/value pairs, which will be written in square brackets. Thanks to the constraint on the relation of values to labels in *FS*, it is possible, in compact notation, to omit labels.

A system CAT specifies no *cat* with $[< l = v_i >, < l = v_j >]$; i.e. in no *cat* specified by system CAT there are two label/value pairs with the same label and different values.

We know that a system CAT specifies the set *MC* ($MC \subset C$), i.e. the set of maximum categories. A *mc* is a category to which, given *FS* and *IR*, no label/value pair can be added by some *Ir* $\in$ *IR*. One or more *mc* are associated to each *lf*.

Different kinds of objects can be elements of *pRSEM*. The basic ambition of 5P related to semantics is to be able to associate any semantic representation to the string's syntax defined in terms of P2 including Arrowing Properties, system CAT and the association in *LE* of *mc* to *lf*. Semantic functions are intended to specify the association of semantic representation to the syntax of the string. The association of some *prsem* to a *lf* is obtained by a subset of Semantic functions. This general goal implies that a *prsem* associated as a final result to a string can be expressed within different *prsem* syntactic notations involving different denotations. E.g. given the string in the following (8.i), the idea is that different Semantic functions must be able to associate either (8.ii) - a first order formula - or (8.iii) - a montaguian one- to one and the same model specified in syntactic terms.

(8)

i Peter runs

ii run'(peter)

iii $[\lambda x[^{\vee}x[^{\wedge}Peter]]][^{\vee}run']$

Because the *rsem* associated to a string of *lf* is built from the *prsem* associated to each *lf* in the string, it is apparent that in a LE, the *prsem's* in *pRSEM* will differ in terms of the final *rsem* which is desired to be obtained. E.g. if a representation in terms of Montague's intensional logic is wanted, the proper noun *Peter* will be associated to a *prsem* of the form $\lambda x[^{\vee}x[^{\wedge}Peter]]$. Those who prefer first order logic will associate to *Peter* some constant *peter*. Those who wish to regularise the representation of common and proper noun, can use something as *peter'(x)*[8].

A *LE* is basically a set of entries. Given the three sets *LF, pRSEM, C* above characterised, it is possible to define two kinds of entries - i.e. elements of a *LE* - from which two different kinds of *LE's* can be defined. The two kinds of entries are defined by the following (9.i) and (9.ii).

(9) Given *LF, pRSEM, C*

i *entry*$_{mc}$ is a pair *<lf, mc>*, such that $lf \in LF$ and $mc \in MC$;

ii *entry*$_{prsem}$ is a triplet *<lf, mc, prsem>*, such that $lf \in LF$, $mc \in MC$ and $prsem \in pRSEM$.

[8] See in Section 9.5 a quite succinct presentation of ongoing work on semantic representations.

Thus *entries*$_{prsem}$ add some information to *entries*$_{mc}$. In parallel to kinds of entries, it is possible to define two kinds of lexicons: *LE*$_{mc}$ and *LE*$_{prsem}$. Given *LF, pRSEM, C* they are, respectively, the set of all and only *entries*$_{mc}$ and *entries*$_{prsem}$.

5 Types of models and granularity of information

Because we want to practice linguistics as if it was indeed an empirical science, we adopted the basic 5P methodological tenet saying that language is not a given object but something that must be studied in terms of chosen points of view. From this it follows that models can be more or less sophisticated, i.e. that the granularity of informations specified in models and their type can change from one description to another in terms of the kind of observations we want to account for, the different kinds of information to be accounted for being represented in different kinds of Protocoles (P1).

One mean to express different levels of sophistication in 5P is the expression or not of Arrowing Properties and the construction or not of some kind of *rsem* from sets of arrowing pairs. So we can distinguish three kinds of models:

1. Basic models;
2. Arrowed models;
3. Semantically described models.

These three kinds of models are organized in a hierarchy, in which any step incorporates the information of the previous one and add some new information. The (2) level can be understood as adding arrows to the (1) level, and the (3) level as adding a semantic representation to the (2) level.

We do nothing in this document with respect to Semantically described models. We sketch here some further discriminations concerning Basic and Arrowed models, and clarify succinctly the relations between model strings and entries in *LE*$_{mc}$ and *LE*$_{prsem}$.

We know that in the model string of a pre-terminal model (see Section 3 in fine) there is no *Sm*; thus the Model substitution rule cannot be applied. Futhermore, all *Sy's* in the model string of a pre-terminal model come from *C*. E.g. (5.i), (5.ii) and (5.iii) are pre-terminal models, and (5.iv) and (5.v) are not pre-terminal models.

Given *MC*, each *cat* in a model string such that $cat \in C \backslash MC$ will be included or, more frequently, properly included in one or more *mc's* in *MC*. If we assume that *[n, f, pl]* and *[n, m, pl]* are *mc* (in which *n* is the value for noun, *f* for feminin, *m* for masculin and *pl* for plural), the *cat* of symbol n_2 of (5.i) - i.e. *[n]* - is properly included in them. Given *MC* it is thus possible to calculate terminal models from pre-terminal ones: in a *terminal model*, all *Sy's* in the model string are *mc*.

Symbols *mc* in model strings allow the linking of models with *LE* entries, either *entries*$_{mc}$ or *entries*$_{prsem}$: entries in *LE* substitute for *mc's* in model strings. Some <*lf, mc*$_j$> (of a *LE*$_{mc}$) or some <*lf, mc*$_j$*, prsem*> (of a *LE*$_{prsem}$) substitutes for a *mc*$_i$ in a model string iff *mc*$_i$ = *mc*$_j$. E.g.: assuming <*fleurs, [n,f,pl]*> is an *entry*$_{mc}$, in the terminal model string which must be associated to (5.i), it will substitute for the *mc* noted *[n, f, pl]*. When <*lf*, mc_j>*'s* are the substitutes, Categorially lexicalised models are obtained; when the substitutes are <*lf*, mc_j, *prsem*>*'s*, Semantically lexicalised models are obtained.

Besides the symbols in *C* (i.e. categories, maximum or not), we assume the set *SM*, i.e. the set of model identifiers of nuclear or non nuclear Properties. Immediately below we sum up distinctions among models *M-Sm* coming from the use of symbols from *C* and/or *SM*.

(10)

i Models obtained without Substitution rules (*M'-Sm*); a symbol *Sy* in the model string of an *m'-Sm* is either a *Sm*, i.e. an $sm \in SM$, or a $cat \in C \backslash MC$, or a $mc \in MC$.

ii Pre-terminal models (*pTM-Sm*), with only *cat's* in model strings; a symbol *Sy* in the model string of *ptm'-Sm* is either a $cat \in C \backslash MC$, or a $mc \in MC$.

ii Terminal models (*TM-Sm*), with only *mc's* in model strings: a symbol *Sy* in the model string of *tm'-Sm* is a $mc \in MC$.

iii Categorially lexicalised models (*L_cM-Sm*), with only *entries*$-mc$ in model strings.

iv Semantically lexicalised models (*L_{sem}M*), with only *entries*$-prsem$ in model strings.

This enables us to refer to *M-Sm* discriminatively when needed: e.g. to *pTM-Sm*, i.e. to preterminal models satisfying Properties-Sm, or to *M'-Sm*, models obtained without the Substitution rules, etc.

Basic models cross with types (10.i) to (10.iv), Arrowed models with types (10.i) to (10.v). Arrowed semantically lexicalised models (i.e. the crossing of Arrowed models with type (10.v)) are the input to the Semantic functions[9].

6 The formal expression of Properties

The final target of 5P is to define Properties(P2) from which it will be possible to obtain semantically specified models. But very little is said in this document on this final issue (See Section 9.5). We present in this Section 6 and in the following two, the basics of P2 formalism we are working on. When the formalism is illustrated by examples, we do not claim that things ARE as they are presented. Presentations illustrate one way of describing things, other alternative ways of doing them within the same formalism certainly exist, and we are not interested in spelling out criteria allowing us to select between alternative formulated Properties. There will be little or none explicative linguistic theory in this paper: we think in general that it belongs to Projections(P3) or Principles(P4), and not to P2 (see some steps in this direction in Section 9.6 below).

Even if the point is absolutely well known by anyone acquainted with the basics of scientific methodology of empirical sciences, we emphasize that there is no inductive process allowing to extract P3 from P2 and/or resulting in the formulation of P4 when "many" P3 have already been described, and, even less, allowing to extract the "good" P2 from the observation of a very big set of Protocoles(P1).

[9] Basic semantically lexicalised models (i.e. Basic models crossing with (10.v)) seem of little interest, because in the absence of Arrowing pairs nothing can be calculated from their *prsem*.

Two different axiomatic systems were distinguished (see Section 3 *in fine*): with and without Substitution rules. We know (see Section 3 *in fine* and Section 5) that *M'-Sm* is the notation for the set of models specified without substitution rules. Given Properties-Sm *Immediate satisfaction* is defined as the conditions to be met by expressions obtained without substitution rules in order to be considered models of Properties-Sm. We reserve *Mediate satisfaction* to refer to the satisfaction of expressions obtained with Substitution rules. In this document we concentrate on immediate satisfaction, that is on *M'-Sm*.

We set up definitions[10] which we intend to be immediately and intuitively understood, but which can be translated directly into a completely formalised language. So we do not use the quantifiers and implication or conjunction symbols; instead we use directly interpretable language expressions (*for all, there exists, if, then* ...).

We use the following notations. Be *S,* S^i, S^j*, ...* sets whose members are symbols *Sy* (i.e. *cat's, IDn's* or *ID's*). Be $\mathcal{A}, \mathcal{Z}$ sets whose members are sets *Ar* (sets of Arrowing pairs, see Section 6.3). Moreover we use the symbol $\sqsubseteq$ and its negation ($\not\sqsubseteq$) in order to express subsumption and its absence between *cat's* and *Sm's*, and the symbols $\overset{\star}{\sqsubseteq}$, $\overset{\star}{\not\sqsubseteq}$ to express subsumption and its absence between *S's*.

We know that a *cat* is a set of label/value pairs (or a set of values in compact notation) and *cat's* can thus be related by inclusion. It is thus possible to define subsumption between *cat's* by the following:

- $cat^i \sqsubseteq cat^j$ if $cat^i \subseteq cat^i$

The subsumption between *Sm's* is defined by the following :

- $Sm^i \sqsubseteq Sm^j$ if $M - Sm^j \subseteq M - Sm^i$

Subsumption and its absence between *S's* (i.e. between sets of symbols Sy) are respectively designated by $\overset{\star}{\sqsubseteq}$ and $\overset{\star}{\not\sqsubseteq}$. We say that:

- $S^i \overset{\star}{\sqsubseteq} S^j$ if there exists a bijective function *f-sub* $S^i \rightarrow S^j$ such as for each symbol $Sy^i \in S^i$ there is one and only one symbol $Sy^j \in S^j$ such that $Sy^i \sqsubseteq Sy^j$ and for each symbol $Sy^j \in S^j$ there is one and only one symbol $Sy^i \in S^i$ such that $Sy^i \sqsubseteq Sy^j$.

Sets other than *cat's* are defined by the following:

S = { $X_1, ..., X_n$ }

S = { x | ... }

When no confusion can arise, singleton sets will be written without curly brackets. Categories will be written in square brackets only by their values, but square brackets of singleton categories will be omitted when no confusion can arise.

[10] The presentation of Properties P2 follows the one already given in [Bès 99a] but with a different notation. The present delayed version benefits from some comments of Thomas Pfuhl to an earlier version of December 99 and from detailed comments and suggestions of Luisa Coheur, who helped us significantly in the improving of the present and last version of our document; as always, the final responsability is entirely ours.

We recall that the general form of a model string model is

$(Sy_1 \ldots Sy_n)$

Positions in a string model will be designated by indexed letters. Letters *i, j, k, l* must be understood as standing for relative positions expressed in alphabetical order; for example *i* designates some position before *j*, but not necessarily the immediate one. Adjacent positions to *i* can be designated *i+1* or *i-1*. Letters *p, q* will be used to designate arbitrary positions, i.e. not necessary relative ones; for example *p* can stand for some position before *q* or after *q*, or the same position as *q*. The expression *p in m'-Sm* is an abbreviation of the expression *the position p in the model string of model m'-Sm*. A symbol *Sy* in some position *p* of a model string is designated Sy_p . We will say that Sy_p *is in a string model* and we designate it $Sy_{p/m'-Sm}$. There is a unique distinguished position in each model string: it is the position filled by the nucleus, the notation of which is $^\circ Sy_p$.

The *pack of a m'-Sm* is defined as follows:

$Pa_{m'-Sm} = \{Sy^i \mid Sy^i = Sy^j$ for all *p* in *m'-Sm* with $Sy^j_p\}$

In $Pa_{m'-Sm}$, which is a set, there is no repetition of the same symbol. If in position *p* and in position *q* in *m'-Sm*, $p \neq q$, there is the same symbol *Sy* (i.e. if we have *Sy*$_p$ and *Sy*$_q$) there is one and the same symbol *Sy* in $Pa_{m'-Sm}$: the objects *Sy*$_p$ and *Sy* are different objects.

Given *M'-Sm*, $\mathcal{P} - Sm$ - the set of all *Pa's* - is defined as follows:

$\mathcal{P}$*-Sm* = $\{Pa^i \mid Pa^i = Pa^j$ for all *m'-Sm* with $Pa^j_{m'-Sm}\}$

Each Property is defined from the four following different points of view:

1. `Notation`; it defines the form of the formula expressing the Property.
2. `Semantic conditions`; it defines the conditions that must be met by the symbols used in the formula ; they are stated in terms of subsumption relations beteen symbols in each formula or with respect to other symbols in other formula of some set of Properties.
3. `Semantics`; it defines the denotational meaning of the formula.
4. `Immediate satisfaction`; it defines the conditions to be met by any expression in order to be considered as a model satisfying the Property; i.e. any model defined by the Property meets these conditions; this view is a corollary from the two previous ones.

Properties are expressed in terms of symbols *Sy's*, i.e. symbols *Sm's* or *cat's*. A system CAT is thus assumed, which specifies, among others, the *cat's* used in the specification of Properties. Futhermore, if a *Sm'* is used in the specification of some of Property *Sm* (i.e. a Property in the set *Pr-Sm*), its associated *Pr-Sm'* is assumed, i.e. the set of Properties which must specify *M-Sm'*.

Properties are intended to behave as analogs of axioms. Thus they are intended to satisfy at least coherence and completeness. From the former, we ask from Properties that no object can be considered both as satisfying and not satisfying them. From the latter, that no object must be in the impossibility of satisfying two of them. But we are not searching, at least at the moment, to prove of the independency of some Property with respect to the others: redundancy, at least if it does not disturb the understanding of the system, is admitted.

Furthermore, we are strongly interested in partial models : objects satisfying some but not all the Properties. This is in order to account for goals stated in Section 1: given some candidate for recognition as a model, we want to be able to state for each Property, if it is satisfied or not by the candidate.

In the following definitions, `Semantic conditions` are stated with the above goals in mind. Some of them are intended to reduce the more crude redundancy: this is why each Property requires to operate with symbols subsumed by symbols in a particular set - i.e. V_{Sm} - defined by the VOCABULARY PROPERTY. Instead, other `Semantic conditions` are intended to ensure completeness inasmuch as it concerns a specific formula, and not a subset of several formulae. For example the `Semantic conditions` of the EXCLUSION PROPERTY are stated in order to keep clear of the possibility of requiring from two symbols Sy^i and Sy^j the exclusion of one by the other if one subsumes the other.

In the subsequent Subsections of Section 6, Existence Properties are presented, while Sections 7 and 8 present Linearity Properties and Arrowing Properties, respectively. Each Property is introduced intuitively in parallel to its more formal presentation.

6.1 Existence Properties

Existence Properties specify sets of symbols, *packs*. They are formally founded on very simple relations of sets: set membership and set inclusions. There is no overt negation operator in formula expressing Existence Properties, although there is a hidden one in EXCLUSION PROPERTY (see Section 6.1.4) and UNICITY PROPERTY; NUCLEUS PROPERTY (see Sections 6.1.2 and 6.1.3), or *NIL* (see Section 6.1.6) can be also expressed by negation.

We distinguish five kinds of Existence Properties: VOCABULARY PROPERTY, UNICITY PROPERTY, NUCLEUS PROPERTY, EXIGENCY PROPERTY, EXCLUSION PROPERTY. We hope that Existence Properties as presented below, despite or thanks to the chosen formal machinery for doing so, can be intuitively and immediately understood. We insist on the idea that they are thought in terms of poor formal relations: the basics of boolean relations. All is expressed in terms of things coming into and going out of sets. In general: the Vocabulary formula allows the building of any string with symbols subsumed by some symbol in*V*; Nucleus and Exigency formulae inject requirements on what must be there; Unicity and Exclusion formulae inject requirements on what must not be there.

The five kinds of Existence Properties are presented in the next five Subsections 6.1.1 to 6.1.5; in 6.1.6 we add several extensions to the definitions presented before.

6.1.1 Vocabulary Property

The VOCABULARY PROPERTY says simply: all symbols in model strings are subsumed by some symbol in *V*. So if, for example, it is wanted to describe nominal nuclear phrases in French, Spanish, Portuguese or English, there must not be in *V* symbols subsuming prepositions, conjonctions or complementizers, but rather categories subsuming articles, proper nouns, common nouns, adjectives, demontratives, possesives and so on.

VOCABULARY PROPERTY

- `Notation`. The unique formula expressing the Vocabulary Property is of the form:
  V_{Sm} = $\{Sy^1, ... , Sy^n\}$

- `Semantic conditions`

  If $Sy^i, Sy^j \in V_{Sm}$, then $Sy^i \not\sqsubseteq Sy^j$.

- `Semantics`

  ◇ For all $Pa_{m'-Sm} \in \mathcal{P} - Sm$ and for all $Sy^j \in Pa_{m'-Sm}$ there exists one and only one $Sy^i \in V_{Sm}$ such that $Sy^i \sqsubseteq Sy^j$.

  ◇ For all $Sy^i \in V_{Sm}$ there exists a $Pa_{m'-Sm} \in \mathcal{P} - Sm$ such that $Sy^j \in Pa_{m'-Sm}$ and $Sy^i \sqsubseteq Sy^j$.

- `Immediate satisfaction`

  m'-Sm satisfies V_{Sm} if for all $Sy^j \in Pa_{m'-Sm}$ there exists a $Sy^i \in V_{Sm}$ such that $Sy^i \sqsubseteq Sy^j$.

Gloss: each symbol in the model string of a *m'-Sm* is subsumed by some symbol in V_{Sm}, and each symbol in V_{Sm} subsumes some symbol in the model string of some *m'-Sm*.

6.1.2 Unicity Property

The UNICITY PROPERTY says simply: if there is in a model string one symbol subsumed by some symbol in *Un*, then, in the same model string, there is no other symbol subsumed by the same symbol in *Un*. This, combined with some judicious organisation of system CAT, allows the expression of some of the paradigmatic relations of structural linguistics, which, despite being scorned from the beginning of chomskyan linguistics, are fundamental to the study of natural languages. For example, suppose some `specif` feature value in French system CAT, subsuming demonstratives, definite articles, indefinite articles, possesives such as *mes* and what we note here as *de'*, the particle used in *de belles fleurs*. It can be easily shown that these five entities must be distinguished as different objects in order to describe French nominal nuclear phrases, but that they exclude each other. This can be expressed by `specif` in Un_{Vn} (see immediately below).

UNICITY PROPERTY

- `Notation`. The unique formula expressing the Unicity Property is of the form:

  $Un_{Sm} = \{Sy^1, \ldots , Sy^n\}$

- `Semantic conditions`

  ◇ If $Sy^j \in Un_{Sm}$, then there is one and only one $Sy^i \in V_{Sm}$ such that $Sy^j \sqsubseteq Sy^k$.

  ◇ If $Sy^j, Sy^k \in Un_{Sm}$, then $Sy^i \not\sqsubseteq Sy^j$.

- `Semantics`

  ◇ For all *m'-Sm* with symbols Sy^j_p, Sy^k_q ($p \neq q$), if $Sy^i \in Un_{Sm}$, then if $Sy^i \sqsubseteq Sy^j$, then $Sy^i \not\sqsubseteq Sy^k$.

- `Immediate satisfaction`

  m'-Sm satisfies Un_{Sm} if for all $Sy^j_{p/m'-Sm}$ such that $Sy^i \sqsubseteq Sy^j$, $Sy^i \in Un_{Sm}$, there exists no $Sy^k_{q/m'-Sm}$, $p \neq q$, such that $Sy^i \sqsubseteq Sy^k$.

Gloss: there are no two symbols in the model string of a *m'-Sm* subsumed by one and the same symbol in Un_{Sm}.

6.1.3 Nucleus Property

The NUCLEUS PROPERTY points out a specific position in a model, occupied by the distinguished symbol $^\circ Sy$. No onthological linguistic characteristics are adjudged neither to the symbol nor to the position in order to advocate for the naturalness of some choice on position and/or on symbol. In 5P, it is simply a chosen unique position which must be occupied by one among choiced symbols in the Nucleus Property formula. It happens that, for example, in French, Spanish and Portuguese, it is appropriate to have nouns and verbs as the nucleus of respective nouns and verb nuclear phrases. But in 5P there is no requirement on feature percolation or the like. Observe in the definitions below that only one of the elements in Nu_{Sm} is required, and that it is not required for noun phrases to have noun categories as the nucleus. So it is possible to express that, for example in French, either some adverbs, adjectives, common nouns or even nuclear prepositional phrases, among others, can be the nucleus of noun nuclear phrases, as illustrated by the following examples.

- ($^\circ Beaucoup_{adv}$ $)_{Nn}$ connaissent la question.
- (Le $^\circ garcon_n$ $)_{Nn}$ connaît la question.
- ($^\circ Pierre_{pn}$ $)_{Nn}$ connaît la question.
- ($^\circ Entre\ quatre\ et\ cinq_{Pn}$ $)_{Nn}$ connaissent la question.
- (Les $^\circ ambitieuses_{adj}$ $)_{Nn}$ connaissent la question.

So there is no need to say that a set of linguistic forms exists in which there are adverbs and indefinite adjectives or pronouns, or to say that any noun is an adjective in order to account for nouns in apposition, or the other way round, that any adjective is a noun in order to account for adjectives - with or without anaphora relations - in noun phrases without some noun, or to say nothing about 'curious' cardinals as *entre quatre et cinq* which as far it can be seen, differ little in behavior with respect to 'ordinary' cardinals (*trois, quatre, cinq, ...*), i.e. they can work as the nucleus or as modifiers in noun nuclear French phrases.

NUCLEUS PROPERTY

- Notation. The unique formula expressing the Nucleus Property is of the form:
  Nu_{Sm} = $\{Sy^1$, ... ,$Sy^n\}$
- Semantic conditions
  ◇ If $Sy^j \in Nu_{Sm}$, then there is one and only one $Sy^i \in V_{Sm}$ such that $Sy^i \sqsubseteq Sy^j$.
  ◇ If $Sy^j, Sy^k \in Nu_{Sm}$, then $Sy^j \not\sqsubseteq Sy^k$.
- Semantics
  ◇ For all *m'-Sm* there exists one and only one position *p* such that $^\circ Sy^j_p, Sy^i \sqsubseteq Sy^j, Sy^i \in Nu_{Sm}$
  ◇ For all $Sy^i \in Nu_{Sm}$
  there exists a *m'-Sm* with an unique $^\circ Sy^j_p$, such that $Sy^i \sqsubseteq Sy^j$.
- Immediate satisfaction
  m'-Sm satisfies Nu_{Sm} if there exists one and only one $^\circ Sy^j_p$ in *m'-Sm* , such that $Sy^i \sqsubseteq Sy^j, Sy^i \in Nu_{Sm}$.

Gloss: in each model string there is one and only one position with a Nucleus symbol which must be subsumed by some symbol in Nu_{Sm}.

6.1.4 Exigency Property

The EXIGENCY PROPERTY says simply: if in a model string there are one or several symbols of certain kind, then there must be in the same model string some other specified symbol(s). Exigency Property formulae inject more requirements than the ones introduced by the VOCABULARY PROPERTY or the NUCLEUS PROPERTY. For instance, the already introduced particle *de'* requires an adjective in some French nominal nuclear phrases; in general, participle verb forms require an inflected auxiliary; in Portuguese of Portugal a possesive requires a definite article or a demonstratif, while an analogous requirement is only valid in French for possessives such as *miens* but not for possessives such as *mes*.

Assuming that *[v,p]* designates participle verb forms and *[aux,f]* auxiliary inflected forms, and assuming *Pr-Vn* (i.e. Properties of the inflected verbal nuclear French phrase), the following Exigency formula expresses the requirement of an auxiliary by the participles:

[v, p] $\Rightarrow_{Vn}$ { . . . [aux, f] . . . }

EXIGENCY PROPERTY

- `Notation`. An exigency formula from the set of exigency formula, is of the form:
  $S^0 \Rightarrow_{Sm} \{S^1, ..., S^n\}$
- `Semantic conditions`
  ⋄ If $Sy^j \in S^{j(j \geq 0)}$, then there exists one and only one $Sy^i \in V_{Sm}$ such that $Sy^i \sqsubseteq Sy^j$ (each *Sy* in a set of the formula is subsumed by one symbol in V_{Sm}).
  ⋄ If $Sy^j \in S^{j(j \geq 0)}$, $Sy^k \in S^{k(k \geq 0)}$, then $Sy^j \not\sqsubseteq Sy^k$ (no *Sy* in a set of the formula subsumes other *Sy* either in the same set or in other set of the formula).
- `Semantics`
  ⋄ For all $Pa_{m'-Sm} \in \mathcal{P}$-Sm
  if $S^0 \overset{\star}{\sqsubseteq} S^x$ and $S^x \subset Pa_{m'-Sm}$,
  then there exists at least one S^k such that
  $S^k \subset Pa_{m'-Sm}$ such that there is some $S^{l(l \geq 1)}$ such that $S^l \overset{\star}{\sqsubseteq} S^k$.
- `Immediate satisfaction`
  m'-Sm satisfies the set of Exigency formula if it satisfies each Exigency formula,
  m'-Sm satisfies an Exigency formula if, given $S^0 \overset{\star}{\sqsubseteq} S^x$,
  ⋄ either $S^x \not\subseteq Pa_{m'-Sm}$
  ⋄ or there exists at least one $S^k \subset Pa_{m'-Sm}$ such that there is some $S^{l(l \geq 1)}$ such that $S^l \overset{\star}{\sqsubseteq} S^k$.

Gloss: in each model string if there is a set of symbols subsumed by the set to the left of $\Rightarrow_{Sm}$ in the Exigency formula, there must be also at least one set of symbols subsumed by one set the the right of $\Rightarrow_{Sm}$.

6.1.5 Exclusion Property

The EXCLUSION PROPERTY says simply: if in a model string there are one or several symbols of certain kind, then there must not be in the same model string such and such other symbol(s). Exclusion Property formulae refine the requirements on symbols in model strings. For instance, indefinite articles (*[art,i]*) and cardinals (*[card]*) cannot be in the same model string of French noun nuclear phrases. This can be spelled out by the following Exclusion formula:

[art, i] $\not\Leftrightarrow_{Vn}$ card

EXCLUSION PROPERTY

- `Notation`. An Exclusion formula from the set of Exclusion formulae is of the form:

  $S^0 \not\Leftrightarrow_{Sm} \{S^1, ..., S^n\}$

- `Semantic conditions`

  $\diamond$ If $Sy^j \in S^{j(j \geq 0)}$, then there exists one and only one $Sy^i \in V_{Sm}$ such that $Sy^i \sqsubseteq Sy^j$ (each *Sy* in a set of the formula is subsumed by one symbol in V_{Sm}).

  $\diamond$ If $Sy^j \in S^{j(j \geq 0)}$, $Sy^k \in S^{k(k \geq 0)}$, then $Sy^j \not\sqsubseteq Sy^k$ (no *Sy* in a set of the formula subsumes other *Sy* either in the same set or in another set of the formula).

- `Semantics`

  $\diamond$ For all $Pa_{m'-Sm} \in \mathcal{P}$-Sm

  if $S^0 \overset{\star}{\sqsubseteq} S^x$ and $S^x \subset Pa_{m'-Sm}$,

  then there is no S^k such that

  $S^k \subset Pa_{m'-Sm}$ such that there is some $S^{l(l \geq 1)}$ such that $S^l \overset{\star}{\sqsubseteq} S^k$.

- `Immediate satisfaction`

  m'-Sm satisfies the set of Exclusion formulae if it satisfies each Exclusion formula,

  m'-Sm satisfies an Exclusion formula if, given $S^0 \overset{\star}{\sqsubseteq} S^x$,

  $\diamond$ either $S^x \not\subseteq Pa_{m'-Sm}$

  $\diamond$ or there is no $S^k \subset Pa_{m'-Sm}$ such that there is some $S^{l(l \geq 1)}$ such that $S^l \overset{\star}{\sqsubseteq} S^k$.

Gloss: in each model string if there is a set of symbols subsumed by the set to the left of $\not\Leftrightarrow_{Sm}$ in the Exclusion formula, there is no set of symbols subsumed by one set to the right of $\not\Leftrightarrow_{Sm}$.

6.1.6 Extensions

The above definitions are intended to express the essentials of the expressive power of Existence Properties. The definitions already given can be extended by the following three notations which, different from the previous ones, add expressive power, while not requiring any extension of the denotational domain within which formulae are evaluated : in all cases the denotations are strings of symbols, and, in all cases the same kind of models are intended to satisfy the extended notations.

The three notations are[11]:

[11] To which it is possible to add a variable onto the nuclear *Sy* required by the NUCLEAR PROPERTY.

1. +S | -S
2. NIL
3. AGR [EEMENT]

Consider the *+S | -S* notation. As it was said in Section 1, we assume that the actual challenge for linguistic descriptions is to be able to account for the fluidity of observations, which express borderline observations.

This challenge meets what seems to be a deep characteristic of human languages. When we have some set of strings in the language, say Str^i, it is possible to express either relations between entities within strings of Str^i or relations between Str^i and some other Str^j saying simply *the relations work for all the strings in Str*^{i} *inasmuch as they have such and such symbol(s) Sy* or *inasmuch as they do not have such and such symbol(s) Sy.*

Within the framework of Existence Properties, this possibility can be expressed directly. If we want to express that we are interested only by some subset of *M'-Sm*, it is possible to define it using the following notation.

$$Sm(+S| - S)$$

+S is the set of symbols *Sy* which are required to be in $Pa_{m'-Sm(+S|)}$ for any *m'-Sm(+S|)*. The other way round, *-S* is the set of symbols *Sy* which are required not to be in $Pa_{m'-Sm(|-S)}$ for any *m'-Sm(|-S)*. Symbols in *+S* or in *-S* in *Sm(+S|-S)* and in linearity formulae (see Section 6.2 below) are of the *Sy* type, i.e. they do not have any relative position associated to them. In Arrowing formulae, symbols in *+S* or in *-S* (see Section 6.3 below), are of the Sy_i type.

For example, given French *Vn* (French verbal nuclear phrase) and French *[cl, nom]* (category subsuming all nominative clitics, as *-je, -il, ...*), and *Vn([cl,nom]|)*, we define *M'-Vn([cl,nom]|)*, i.e. the set of models with a nominative clitic (as *a-t-il regardé*). Instead, with *Vn(|[cl,nom])*, we define *M'-Vn(|cln)*, i.e. the set of models without a nominative clitic (as *la regarde*). The two notational conventions can be composed. For example *Vn([cl,nom]|aux)* defines *M'-Vn([cl,nom]|aux)*, i.e. the set of models with a nominative clitic and without an auxiliary.

These notational conventions change nothing to the above definitions of Existence Properties. Furthermore, given the already presented definition on subsumption of *Sm's* symbols, we have that

$Sm \sqsubseteq Sm(+S| - S)$

This is directly true for *-S*. In this situation we have *M'-Sm(<|-S)* $\subset$ *M'-Sm* . When we have *+S* the same proper inclusion is obtained, with the exception of some borderline situations (for example, the one araising when *+S* = Nu_{Sm}, when $Nu_{Sm} = \{Sy\}$). But in these we also have *M'-Sm(+S|)* $\subseteq$ *M'-Sm*.

The *NIL* notation introduces a hidden negation. It's use is strictly reserved to the right of Exigency formula. We have the two following notations of formula with *NIL* (Semantic conditions related to $S^{i(i \leq 0)}$ are the same than those for Exigency formula without *NIL*).

i $S^0 \Rightarrow_{Sm} NIL$

ii $S^0 \Rightarrow_{Sm} \{NIL, S^1, ..., S^n\}$

Formula (i) expresses:

$\diamond$ For all $Pa_{m'-Sm} \in \mathcal{P}$-Sm

if $S^0 \overset{\star}{\sqsubseteq} S^x$, then $Pa_{m'-Sm} = S^x$.

Formula (ii) expresses:

$\diamond$ For all $Pa_{m'-Sm} \in \mathcal{P}$-Sm

if $S^0 \overset{\star}{\sqsubseteq} S^x$, then

either $Pa_{m'-Sm} = S^x$,

or there exists at least one S^k such that $S^k \subset Pa_{m'-Sm}$ and there is some $S^{l(\geq 1)}$ such that $S^l \overset{\star}{\sqsubseteq} S^k$.

Formula (i) can be used to express, for example, that personal pronouns (which we note *pprn*) in French, English, Spanish and Portguese (e.g. French *je, tu*, ...; English *you*, etc.) are the unique forms admitted in the nuclear nominal phrase; we write:

pprn $\Rightarrow_{Nn}$ NIL

Formula (ii) can be used to express, for example, that plural nouns in Spanish or English can be used either as unique forms admitted in the nuclear nominal phrase, or with some determinant.

The use of *NIL* is a shorthand notation to account for linguistic attested situations which can be accounted for without it, but in a more cumbersome way. The example given below of personal pronouns, assuming:

- $V_{Nn} = \{Sy^1, \ldots, Sy^n\}$;
- $Sy^1 \sqsubseteq pprn$;
- $x^1, \ldots, x^n$ all other *Sy's* different from *pprn*, not subsumed by *pprn*, not subsuming each other and subsumed by Sy^1;
- $Sy^1 \notin Un_{Sm}$;

can be equivalently expressed by the following Exclusion formula

$pprn \not\Leftrightarrow_{Nn} \{x^1, ..., x^n, Sy^2, \ldots, S^n\}$

That is, instead of saying in an Exigency formula that *pprn* requires *NIL*, we can say, in an Exclusion formula, that it rejects anything which, in terms of V_{Nn} may be in a nominal nuclear pack. Observe that if $Sy^1 \in Un_{Nn}$, then symbols $x^1, \ldots, x^n$ in the Exclusion formula are redundant. The general conjecture, that we do not develop further here, is that any Exigency formula of the types (i) or (ii) above can be equivalently expressed by Exclusion formula(e)[12].

Conceptually, agreement in natural language is the requirement of some set of label/value pairs in some symbol in a model string when some other entity in the same model string has the

[12] For the same reasons but the other way round, it is unnecessary to define an Exclusion formula as $S^0 \not\Leftrightarrow_{Sm} NIL$. For example, if there is at least one Exigency formula $S^0 \Rightarrow_{Sm} X$, the Exclusion formula is redundant. Here also the general conjecture is that Exclusion formulae with *NIL* can be equivalently expressed by Inclusion formulae, and the examined linguistic material does not suggest the need to define them in order to obtain some further syntactic notational shorthand.

same ones. Exigency Properties express in general these kinds of requirements; they say that if there is something in some model string, in the same model string there must be something else. Agreement can thus be expressed directly by Exigency Properties.

We sketch here a possible way of accounting for agreement phenomena, or at least, for some of them.

We suppose a system CAT with a feature set *FS* (see preceding Section 4). It is possible to pick out one or more features in *FS* and define *cat's* with label/value pairs from these features. Suppose we have in *FS* the features:

<num[ber]; {sg[singular], pl[plural]}

<per[son]; {p1, p2, p3}>

<gen[der]; {masc[ulin], fem[inin]}

We accept that a variable can be a possible value of the above three labels. We define *AGR* as a variable on *cat's* such that all their values come from one or more of these three features, and we define *N, P, G* as variables on *cat's* with respectively only one value of the labels *num, per, gen*. We can combine *N, P, G* in order to obtain two or three of these labels, e.g. *NP* (*cat's* with values of *N* and *P*), or *PG* (*cat's* with values of *P* and *G*), etc.

With *X* as a metavariable on *AGR, N, P, G* or on some combination of the last three of them, we will designate in *X* of

$catX$

the set of label/value pairs (or set values in compact notation) instantiating *X* and included in *cat*. E.g., given:

cat = [n[oun], sg, p3, fem]

AGR in the symbol *catAGR* stands for *[sg, p3, f]*, *P* in the symbol *catP* stands for *[p3]*, etc.

For the time being, we allow the use of unification between agreement features. If in a formula we have

$cat^iX \ldots cat^jX$

with instantiation of *X* in both cat^iX and cat^jX by the same *AGR, N, P, G* or by some combination of the last three of them, we express requirement of unification between the involved agreement features. On the other hand, the corresponding notations ¬*AGR*, ¬*N*, ¬*P*, etc. express the requirement of failure of unification between the involved agreement features.

For example, be the following Exigency formula (with curly brackets omitted):

Sy^iAGR $\Rightarrow_{Sm}$ Sy^jAGR

The formula means not only that Sy^i requires an Sy^j in exactly the same terms as in any Exigency formula, but, furthermore, that *Sy's* symbols respectively subsumed by Sy^i and by Sy^j must satisfy unification of values of the *num, per* and *gen* labels.

It is thus possible to express requirements introduced by object clitics in the French verbal nuclear phrase with the following Exigency formula:

[*cl,le*]GN $\Rightarrow_{Vn}$ {[*1v,a,f*], [*1v, a, p*]GN}

In the formula, *[cl, le]* specifies object clitics, *[1v, a]* specifies a class of verbs admitting *avoir* forms for the auxiliary, and *[f]* and *[p]* specify respectively inflected forms, and participle forms. The formula says: *a clitic object requires an inflected verbal form of verbs of the [1v, a] class, or a participle of the same class of verbs agreeing in gender and number with the clitic object.*

It is sometimes cumbersome to express agreement conjointly with ordinary requirements expressed by Exigency formulae. So an ad-hoc Agreement Exigency formula can also be used to do this. It is the following one, labelled *Agreement formula*:

$$S^i a \Rightarrow_{Sm} X$$

An Agreement formula has the following `semantics`.

◇ For all Pa$_{m'-Sm}$,

if cat^i, cat^j $\in$ Pa$_{m'-Sm}$, cat^k, cat^l $\in$ S^i,

such that cat^k $\sqsubseteq$ cat^i, cat^l $\sqsubseteq$ cat^j,

then cat^{i}X, cat^{j}X.

Thus, an Agreement formula means that *cat's* subsumed by *cat's* to the left of a$\Rightarrow_{Sm}$ must agree when used in some model string, which is expressed by the unification of *X*. As in Exigency formulae with the expression of agreement, negative requirements can be expressed in an Agreement formula on *X*. As an example : suppose that the following ill French forms (1) to (3) must be excluded while forms (4) to (6) must be accepted.

1. *Vous t'avez dit que ...
2. *Tu vous as dit ...
3. *Nous m'avons dit ...
4. Tu t'es dit ...
5. Il t'a dit ...
6. Nous nous sommes dit que ...

The following Agreement formula, in which *[a, aux, f]* specifies *inflected avoir auxiliary*, and *[cl, r]* specifies *reflexive clitic*, can be used:

{ [a, aux, f], [cl, r] } a$\Rightarrow_{Vn}$ $\neg$P

7 Linearity Properties

The basic goal of Linearity Properties is to associate order relations to packs. They are defined within the same pattern of presentation as Existence Properties.

LINEARITY PROPERTIES

- `Notation`. A linearity formula from the set of linearity formulae is of the form:
  $Sy^0 \prec_{Sm} Sy^1, ..., Sy^n$

- `Semantic conditions`

  ◇ For $Sy^{i(\geq 0)}$ there is one and only one $Sy^x \in V_{Sm}$ such that $Sy^x \sqsubseteq Sy^i$.

  ◇ For $Sy^{i(\geq 0)}, Sy^{j(\geq 1)}, Sy^i \not\sqsubseteq Sy^j, Sy^j \not\sqsubseteq Sy^i$.

- `Semantics`

  ◇ For all $m' - Sm$ such that $Sy^k_{p/m'Sm}, Sy^l_{q/m'Sm}$,

  if $Sy^0 \sqsubseteq Sy^k, Sy^{i(i \geq 1)} \sqsubseteq Sy^l$,

  then $p = i, q = j$.

- `Immediate satisfaction`

  m'-Sm satisfies the set of Linearity formulae if it satisfies each Linearity formula,

  m'-Sm satisfies a Linearity formula if,

  ◇ either Sy^k, Sy^l as in the previous view (`Semantics`) are not in the model string of *m'-Sm*,

  ◇ or Sy^k_i, Sy^l_j are in the model string of *m'-Sm*.

Gloss: in a model string if there is a symbol subsumed by the symbol to the left of $\prec_{Sm}$ and a symbol subsumed by a symbol to the right, the former precedes the latter one.

By means of the +*S*|-*S* concepts presented in Section 6.1.5, the above Linearity formula can be extended. It becomes:

$Sy^0 \prec_{Sm} Sy^1, ..., Sy^n$

+S^k

-S^l

Thus S^k in +S^k designates a set such that the linearity requirements stated by the formula must be satisfied by *m'-Sm* if there is a set $S^{k'}$ such that $S^k \overset{\star}{\sqsubseteq} S^{k'}$, $S^{k'} \subset Pa_{m'-Sm}$. *Mutatis mutandis* the same is valid with regards to S^l.

Sy's symbols in +S^k or in -S^l are not of the Sy_i type, i.e. they do not have a relative position associated to them.

Note that in Linearity formulae, the symbol to the left of $\prec_{Sm}$ precedes each of the symbols to the right, and that the order of the symbols placed to the right is not significant for those symbols.

As an example: in French nuclear noun phrase (i.e. N_n) the forms *tous/toutes* (specified by [t]) precedes all the others (n[ouns], adj[ectives], dem[ostratives], art[icles], pos[sessives]). This can be expressed by:

[t] $\prec_{Sm}$ [n], [adj], [dem], [art], [pos], [pos]

8 Arrowing Properties

In Section 1 we used two different notations for Arrowed models (cf. (6) and (6b)), as in the following (1) and (2). The general pattern of (2) is the one of (3).

1. $(card_{1\to 3}ADJ1_{2\to 3}n_{3\to 3})_{Nn}$
2. $<(card_1ADJ1_2n_3)_{Nn}, \{<1,3>, <2,3>, <3,3>\}>$
3. $<<$ `model string` $>_{Sm}, <$ `set of Arrowing pairs` $>>$

A set of Arrowing pairs specifies a graph. In Arrowed models, model strings are thus associated to a graph. This graph is in the input of the Semantic functions; it is an important element of Semantically lexicalised models (see Section 5). The basic goal of Arrowing pairs is thus crucial to contribute to the specification of semantic representations.

One of the sources of ambiguity comes from the fact that a unique model string can be associated to more than one set of Arrowing pairs. For example, it is possible to describe the string of linguistic forms in the following (1) with one of the two sets of Arrowing pairs (Ar^i, Ar^j) in (2).

1. $(\text{Pierre})_1$ $(\text{a regardé})_2$ $(\text{les filles})_3$ $(\text{avec des lunettes})_4$
2. Ar^i= {<1,2>, <3,2>, <4,2> }, Ar^j= {<1,2>, <3,2>, <4,3> }

We define $\mathcal{A}$ as the set of sets of Arrowing pairs associated to a string model. So the general form of an Arrowed model is:

$<<$ `model string` $>_{Sm}, \mathcal{A}>$

Arrowing Properties associate appropriate $\mathcal{A}$'s to model strings. *Ar's* are the elements in $\mathcal{A}$'s. Because a graph is basically a set of pairs, restrictions on the set define different types of graphs.

We introduce the following `General semantic conditions` on any *Ar*.

- Each pair in *Ar* is of the form $Sy^i_p \to Sy^j_q$.
- For each Sy^i_p in *m-Sm*, there is one and only one pair $Sy^i_p \to Sy^j_q \in$ Ar.
- There is in *Ar* one and only one pair $Sy^i_p \to Sy^j_q$ with $p = q$.

Furthermore, we require that, ignoring the Root Arrowing pair[13] the graph specified by the set of Arrowing pairs must be an acyclic graph. The above conditions are intended to ensure that a graph defined by an *Ar* is a connected graph with a root, this being the unique *Sy* arrowing to itself. Observe that the crossing of arrows is not excluded by `General semantic conditions` and, furthermore, that arcs connecting *Sy's* in Arrowing pairs are not labelled. Observe also that Sy^i symbols can be of the Sy^i_i type, i.e. with relative positions expressed on them. The underlying - and yet to be tested - assumption is that graphs defined within the above `General semantic conditions` can indeed express the 'connecting' relations between *Sy's* from which semantic representations can be built by the Semantic functions.

Arrowing Properties are defined within the same pattern of presentation as that used for Existence and Linearity Properties; with +S^k and -S^l in a `subformula` we designater symbols which must be there and symbols which must not be there, respectively, as in Linearity Properties; elements in them are of the Sy_i type.

[13] We thank François Trouilleux for the discussion of this point.

- Notation. An Arrowing formula from the set of Arrowing formulae is of the form

  <subformula>$_1$

  ;

  ...

  ;

  <subformula>$_n$

  where ';' expresses disjunction and <subformula>$_i$ is of the form

  $Sy^i_p \rightarrow_{Sm} Sy^j_q$

  +S^k

  -S^l

- Semantic conditions

  ◇ Given a subformula,

  if $p = q$ (and thus $Sy^i_p = Sy^j_q$), then there is one and only one $Sy^k \in V_{Sm}$ such that $Sy^k \sqsubseteq Sy^i$,

  if $p \neq q$, then there is one and only one Sy^k, and one and only one $Sy^l, Sy^k, Sy^l \in V_{Sm}$ such that $Sy^k \sqsubseteq Sy^i, Sy^l \sqsubseteq Sy^j$.

- Semantics

  ◇ For all <*m'-Sm*, $\mathcal{A}$>, <subformula>$_i$
  with $Sy^k_r \in +S^k$, $Sy^l_s \in -S^l$, $Sy^1_{p'}, Sy^2_{q'}, Sy^3_{r'} \in m - Sm$, $Sy^4_{s'} \notin m - Sm$,

  let $Sy^1_{p'}$ be such that $Sy^i_p \sqsubseteq Sy^1_{p'}$;

  let $Sy^2_{q'}$ be such that $Sy^y_q \sqsubseteq Sy^2_{q'}$;

  let $Sy^3_{r'}$ be such that $Sy^k_r \sqsubseteq Sy^3_{r'}$;

  let $Sy^4_{s'}$ be such that $Sy^l_s \sqsubseteq Sy^4_{s'}$.

  If the order of relations between *p, q, r, s* are the same as the order of relations between *p', q', r', s'*,

  then there exists one and only one $Ar \in \mathcal{A}$ such that $< Sy^1_p, Sy^2_q >= Ar$.

- Immediate satisfaction

  ◇ < *m'-Sm*, $\mathcal{A}$ > satisfies the set of Arrowing formulae if

  < *m'-Sm*, $\mathcal{A}$ > satisfies each formula in the set,

  < *m'-Sm*, $\mathcal{A}$ > satisfies a formula if

  < *m'-Sm*, $\mathcal{A}$ > satisfies disjunctively each <subformula> of the formula.

  ◇ For all < *m'-Sm*, $\mathcal{A}$ >, <subformula>$_i$ with symbols

  Sy^i_p, Sy^j_q,Sy^k_r, Sy^l_s, $Sy^1_{p'}$, $Sy^2_{q'}$, Sy^3_r, $Sy^4_{s'}$ as in the previous view (cf. Semantics)

  and with the order of relations between *p, q, r, s* being the same as the order of relations between *p', q', r', s'*,

  < *m'-Sm*, $\mathcal{A}$ > satisfies <subformula>$_i$ if there exists one and only one $Ar \in \mathcal{A}$ such that $< Sy^1_p, Sy^2_q > \in Ar$.

Gloss of a subformula: a symbol in some position in a model string arrows either to itself or to a symbol in some other position, inasmuch as symbols subsumed by symbols in $+S^k$ are attested in the model string, and symbols subsumed by symbols in $-S^l$ are not attested [14].

The following is an illustration of ARROWING PROPERTIES. Consider the three following strings; substrings in brackets are French verbal nuclear examples.

1. Jacques (les a-t-il tous regardés) ...
2. Nous (avons tous regardé) ...
3. Nous (les avons tous regardés) ...

Assume the following informations comes from Protocoles.

(11)

i In (1) *tous* 'semantically specifies' *les*.

ii In (2) *tous* 'semantically specifies' *nous*.

iii In (3) *tous* 'semantically specifies' either *les* or *nous*.

The expression *'semantically specifies'* can be clarified in different ways. The final target is:

> Knowing the maximum categories and the primitive semantic representation (i.e. *prsem*) associated in the lexicon to the items *nous, tous, les*, the Semantic functions, having as input the Semantically lexicalised models associated to the above examples are intended to specify semantic representations which are intended to account for any definition of *'semantically specifies'*.

Accept furthermore that there is some Arrowing pair (not formally expressed in the following) which relates *nous*, outside the verbal nuclear string, to the auxiliary, which is inside. Then the following Arrowing formulae (remember that *[cl,le]* stands for clitic objects with forms *les, la, le, l'*; *pl* stands for plural, and *t* for *tous*) express what is needed:

[t] $\rightarrow_{Vn}$ [a, aux, pl]

;

[t] $\rightarrow_{Vn}$ [cl,le, pl]

The formula, by its first subformula, expresses that in any case, the form *tous* arrows to an *avoir* plural auxiliary, if it happens that *tous* and the auxiliary are in the same model string. By its second subformula, it expresses that if it happens that *tous* and a *le* form are in the same model string, then, within the terms of an exclusive disjonction (the notation of which is ";" and which is derived from the General semantic condition on Arrowing Properties; see above) with respect to the other subformulae, the form *tous* arrows to the clitic *le*; i.e; the ambiguity shown in (11.iii) is expressed.

Suppose that Protocoles, as customary, are not clear with respect to the semantic specifications needed for (3), and that it is wanted to express that in (3) *tous* 'semantically specifies' only the

[14] A more expressive specification of Arrowing Properties can be given if instead of +S^k and -S^l in a `<subformula>` we use $+s\mathcal{A}$, i.e. a subset of Arrowing pairs which must be included in $\mathcal{A}$ of <*m'-Sm*, $\mathcal{A}$>, and $-s\mathcal{A}$, i.e. a subset of Arrowing pairs which must not be included in $\mathcal{A}$.

clitic. The preceding formulae must be changed into the following, where *-[le, pl]* (i.e. the *-S* concept) eliminates the ambiguity of arrowing.

[t] $\rightarrow_{Vn}$ [a,aux,pl]

-[le, pl]

;

[t] $\rightarrow_{Vn}$ [le, pl]

Observe that there is no orde required in subformulae. The previous and the following have the same semantics.

[t] $\rightarrow_{Vn}$ [le, pl]

;

[t] $\rightarrow_{Vn}$ [a, aux, pl]

- [le, pl]

9 Antecedents, published and ongoing work

A general overview of 5P antecedents with pointers to published and ongoing work is presented in the following Subsections: 9.1 Antecedents, 9.2 Complementary work on Properties, 9.3 Processes, 9.4 Descriptive work, 9.5 Semantics, 9.6 Projections. In 9.7, 5P references are listed.

9.1 Antecedents

At the end of the eighties, after working for several years with GPSG and categorial grammars, we were convinced at the GRIL (Groupe de Recherche dans les Industries de la Langue) research team of Blaise-Pascal University, that these kinds of models are in the twofold incapacity to understand natural languages and to be the underlying knowledge source which must be accessed in the computational processing of natural languages, despite the strong and significant improvements they bring to the study and comprehension of natural languages.

The diagnosis was - and it remainds: a grammar, as the concept is understood in GPSG, categorial grammars, LFG, HPSG, TAG, etc. is a hybrid object which is intended to fill two different roles: the one of being a systematic description of linguistic observations and the one of being the declarative source for the processing of algorithms. Futhermore, and on the methodological side of the diagnosis, the idea is that despite many declarations on the contrary, linguistics have never been practiced as an empirical science. So the step subsequent to this twofold diagnosis was to try to dissociate the descriptive function from the procedural one, and to try to define an overall pattern where it should indeed be possible to practice linguistics as an empirical science.

At the end of the eighties (cf. [Bès & Jurie 89] extended in [Jurie & Bès 92]) the distinction was introduced between *descriptive metalanguage* (today *Properties*) and *grammar*; afterwards the descriptive metalanguage became *axioms* in the *A & A* presentation (i.e. *Axioms and Algorithms*, cf. [Bès 93]). But the basis of today'zs 5P Pattern was not defined before [Bès 97a], [Bès 97b], [Bès 97c], [Bès 98] with the label *3P*, covering Protocoles, Proprieties and Processes. The formalism of the Proprieties was essentially the one presented in the preceding Sections, but with a different notation and much less formal explicitness. The discussions with Blache (see footnote 1) in 1998 lead to the enrichment of the *P's* collection which increased to 5.

9.2 Complementary work related to Properties

In the previous Section 4 we assumed a system CAT and in Section 3 *in fine* a Model substitution rule with no or very little formal detail.

System CAT is presented with some detail in [Bès 01a]. Definitions of categories and of inheritance relations were already used in previous descriptive work (cf. [Bès 99a]). The improvement in [Bès 01a] is a better specification of Inheritance relations which can now specify in a compact way a monotonic inheritance system.

A characterisation of four possible types of the Model substitution rule is given in [Bès 99b]. Four possible types because the rule is defined for Basic and Arrowed models, and in each case, with and without symbols satisfying the Unicity Property.

In the previous Sections nothing is said on the expressive power of P2. The issue is tackled in [Bès 99c] where the expressive power of P2 is situated at the level of grammmars of type 1 in the chomskyan hierarchy inasmuch as Arrowing pairs are specified as seen in footnote 13: this paper shows how, with P2, it is possible to specify languages with an indefinite number of *A*'s followed by the same number of *B*'s and *C*'s.

9.3 Processes

A central point of the 5P Paradigm is the dissociation of the descriptive issue from the computational processing one, the last not being necessarily limited to computational parsing.

But dissociation does not mean mutual ignorance. Rather that the challenge is to extract from P2 the information needed to attain a particular computational objective. An important step in this direction is [Hagège 00] where, on the one hand, Portuguese nominal phrase is thoroughly described in terms of P2 (chapters 3 and 4), a function is explicited allowing to extract from categories and P2 the *leaves* (French *feuilles*) and the parser *AF* is defined and implemented (chapter 7): *AF* for French *Analyseur par les Feuilles*, the *leaves* (French *feuilles*) being the basic declarative source of knowledge to which the parser accesses and which are calculated from categories and P2; cf. synthetic previews of [Hagège 00] in [Hagège & Bès 99] and [Bès, Hagège & Coheur 99]. *AF* is also used in [Rodier 00].

In a borderline situation, the declarative source of processing algorithms can be Properties (P2) and system CAT. This is the case of the Model Generator, specified but not implemented in [Bès 99d] and [Bès & Coheur 00]. The challenge for the Model Generator is, given as input P2-X and its associated system CAT, to be able to specify M-X with all and only the models satisfying P2-X, and, in an ideal situation, (not prospected in the actual versions of the Model Generator) to characterise also partial models, i.e. strings satisfying only partially P2-X, see Section 6 *in fine*.

Another method for parsing is considered in [Bès & Blache 99]; see also footnote 1.

9.4 Descriptive work

Detailed descriptive work in the 5P Paradigm has been done in Portuguese and French. For Portuguese see the preceding Section 9.3 where [Hagège 00] is presented. [Bès 99a] is a fairly detailed description of French nuclear verbal phrase, while [Bès 98b] is a rather succinct description of some French comparative descriptions[15].

9.5 Semantics

The idea that Arrowing pairs are an essential element of the input to the Semantic functions which are intended to specify semantic representations is shown in [Bès 99a] and the mechanism for doing so with the Model substitution rule applied to Arrowed models is illustrated in [Bès 99b].

Formal semantics is strongly in debt to Montague's work. [Bès 01b] takes a position on Montague Grammar. Besides a critical analysis of Montague's tenet assimilating formal and natural languages, it justifies the non adoption by the 5P Paradigm of Montague's requirement of syntax/semantics homomorphism. [Bès 00] explores a description of simple French sentences which incorporates Semantic functions. Their output structures - i.e. semantic representations - are inspired by indexed languages, but in the actual presentation they do not incorporate quantifiers. These semantic representations are thus a kind of intermediate representations. They basically incorporate a predicate notation and shared variables; the ',' symbol here has the semantics of &, i.e. the order of the expressions is not significant. The semantic representations can be illustrated by the following examples.

1. Les enfants jouent au ballon dans le jardin.

   jouer'(x0, x1, x2), def'(x3, x1), enfant'(x1), ballon'(x2), dans'(x4, x0, x5), def'(x6, x5), jardin'(x5)

2. Marie regarde la soeur très heureuse de Pierre.

   regarder'(x0, marie, x1), def'(x2, x1), soeur'(x1, pierre), tres'(x3, x4), heureux'(x4, x1)

3. La fille capable de rêver dort tranquillement.

   dormir'(x0, x1), def'(x2, x1), fille'(x1), capable'(x3, x1, x4), rever'(x4, x1), tranquillement(x5, x0)

4. La fille capable de rêver dort très tranquillement.

   dormir'(x0, x1), def'(x2, x1), fille'(x1), capable'(x3, x1, x4), rever'(x4, x1), tranquillement'(x5, x0), tres'(x6, x5)

9.6 Projections and Principles

Projections(P3) are generalisations over Properties (P2) or subsets of Properties of some natural language, and/or over strings described by Properties. Within this general concept several tracks were followed in order to try to approach Projections(P3), or, in the best cases, (modest!) Principles(P4).

[15] The description was used in Blache's document refered to in footnote 1.

The concept of *nuclear phrase* (French *syntagme noyau*) underlies chunk grammars. A nuclear phrase can be characterised as a sequence of categories running from the one detected as being the initial one, to the one that is ordinarily considered as being the head of the phrase. E.g. strings in italics in (1) and (2) immediately below are French nuclear verbal phrases, in (3) and (4) they are French inflected nuclear nominal phrases, and in (5) and (6) they are English nuclear verbal phrases.

1. Il *ne le lui a pas donné*.
2. Il *regarde* la fleur.
3. *Les trois belles fleurs* sont ici.
4. *Pierre* est gentil.
5. He *has been called* by Peter.
6. She *is* nice.

In [Bès 97a] and [Bès 97b] it was pointed out that, with the simplification of ignoring coordination, the sets of strings of the nuclear phrases illustrated before, are *K1F* languages, i.e. finite languages which can be specified by a finite state automaton *K* limited, with *K = 1*. From this, interesting consequences follow which allow us to calculate, from the observation of a reduced corpus of strings of each type of nuclear phrases, the description of the whole corpus.

[Bès 98a] points out that, given some set *S* of categories, any string in the set of strings of, for example, French nominal nuclear phrases, cannot be followed by any *cat* in *S*, even if any *cat* in *S* can follow some nominal nuclear phrase(s). E.g. (1) and (2) are French nominal nuclear phrases, but (1) and not (2) can be followed by a noun in apposition: (3) and (5) are well formed, while (4) is not.

1. (La fille)$_{Nn}$ est partie.
2. (La malheureuse)$_{Nn}$ est partie.
3. (La fille)$_{Nn}$ mère est partie.
4. $*$ (La malheureuse)$_{Nn}$ mère est partie.
5. (La malheureuse mère)$_{Nn}$ est partie.

This generalisation can be extended to many other nuclear phrases in French, Portuguese and Spanish, and even to the detection of Spanish syllabic types, and can thus be ranged as an incipient (but modest!) Principle (P4). It was labelled *diabolic transition* (in French *transition diabolique*) and formally characterised in terms of coding theory: in the previous examples, a string *s* resulting from the concatenation of a string *s'* of a nuclear phrase with an immediate *cat* and preserving the Nucleus in *s'* is not well formed if there is a string *s* which is also a nuclear phrase. That is, strings in the nuclear phrase and strings resulting from concatenation of these with an immediate category obey restrictions of left three codes. Advantage can be taken from this in parsing because, in many cases, given some sequence of immediate categories

$...cat_i cat_j...$

it can be detected if cat_i is the right-hand limit of some nuclear string, cf. [Bès, Hagège & Coheur 99]; the diabolic transition concept was used for Portuguese parsing in [Hagège 00] and for the control of simplified English employed in technical documentation in [Rodier 00][16].

K1F and the diabolic transition are not directly expressed on P2 but on strings of symbols which can be described by P2. A different track was followed in [Bès 99e]. The underlying idea of this paper is to take advantage of the embedding of identical *IDn's* in different *ID's*. E.g. if we have French *IDn Nn* (nominal nuclear phrase), *PREn* (prepositional nuclear phrase), *ADJn* (adjective nuclear phrase), it can be observed, on one hand, that each *IDn* is embedded as the Nucleus in its associated *ID* (i.e., *Nn* is the Nucleus of *N*, *ADJn* the Nucleus of *ADJ* and *PREn* the Nucleus of *PRE*), and that *ADJn* and *PREn* are in the *N* Vocabulary, *PREn* in the *ADJ* Vocabulary and *ADJn* in the *PRE* Vocabulary. The example in the following (1) illustrates an *N* with several embedded *ADJn's* and *PREn's*; (2) and (3) illustrate *ADJ's* embedded in (1); (4), (5) and (6) *PRE's* embedded in (1) (remember that $^\circ Sy$ is the notation for the Nucleus).

1. ((°la machine)$_{Nn}$ (adéquate)$_{ADJn}$ (pour le blanchissage)$_{PREn}$ (du linge)$_{PREn}$ (avec de l'eau)$_{PREn}$ (chaude)$_{ADJn}$)$_{N}$
2. ((°adéquate)$_{ADJn}$ (pour le blanchissage)$_{PREn}$ (du linge)$_{PREn}$ (avec de l'eau)$_{PREn}$ (chaude)$_{ADJn}$)$_{ADJ}$
3. ((°chaude)$_{ADJn}$)$_{ADJ}$
4. ((° pour le blanchissage)$_{PREn}$ (du linge)$_{PREn}$ (avec de l'eau)$_{PREn}$
   (chaude)$_{ADJn}$)$_{PRE}$
5. ((°du linge)$_{PREn}$)$_{PRE}$
6. ((° avec de l'eau)$_{PREn}$ (chaude)$_{ADJn}$)$_{PRE}$

The description of the previous observations requires P2 with the Pr-N, Pr-ADJ, Pr-PRE sketched in the following (12.i), (12.ii) and (12.iii) respectively.

(12)

i V$_{N}$ = {Nn, ADJ, PRE}
Nu$_{N}$ = {Nn}
ADJ$_j$ $\rightarrow_N$ Nn$_i$
PRE$_j$ $\rightarrow_N$ Nn$_i$

ii V$_{ADJ}$ = {ADJn, PRE}
Nu$_{ADJ}$ = {ADJn}
PRE$_j$ $\rightarrow_{ADJ}$ ADJn$_i$

iii V$_{PRE}$ = {PREn, ADJ, PRE}
Nu$_{PRE}$ = {PREn}
ADJ$_j$ $\rightarrow_{PRE}$ PREn$_i$
PRE$_j$ $\rightarrow_{PRE}$ PREn$_i$

[16] The metaphor of diabolic transition thus becomes more clear: in classic musicology, a *diabolus* designates a musical note which cannot follow some characterised sequence of notes.

(12.i) states that the Vocabulary of *N* is a set with *Nn, ADJ* and *PRE* as members, that *Nn* is the Nucleus symbol, and that *ADJ* and *PRE* arrow to a *Nn* on their left. Analogous Proprieties are spelled out in (12.ii) and (12.iii). The interaction of (12.i) with (12.iii) results in a kind of recursivity: a $^\circ Nn$ (i.e. an Nn as Nucleus) can be followed by strings with an indefinite number of*ADJn's* and/or of *PREn's*. Drawing on this, [Bès 99e] proposes a description not in terms of P2 as in (12) but in terms of their Projection(P3) in (13); see a different version in [Bès & Coheur 99]. That is, instead of having three different *Pr-N, Pr-ADJ, Pr-PRE* as in (12i) to (12.iii), (13) compresses the description to *Proj(N, ADJ, PRE)* as in the following (13).

(13) i V_{Proj} = {Nn, ADJn, PREn }

ii Nu_{Proj} = {Nn, ADJn, PREn }

iii $\text{PREn}_j \rightarrow_{Proj} \text{Nn}_i$; PREn_i; ADJn_i

iv $\text{ADJn}_j \rightarrow_{Proj} \text{Nn}_i$; PREn_i

v $^\circ\text{Nn}_i \rightarrow^\circ_{Proj}\text{Nn}_i$

vi $^\circ\text{PREn}_i \rightarrow^\circ_{Proj}\text{PREn}_i$

vii $^\circ\text{ADJn}_i \rightarrow^\circ_{Proj}\text{ADJn}_i$

In [Bès 99e] the function $F(E) = Proj_E$ is defined such that E is a set of which each element is a $Pr - IDn$, and $Proj_E$ is expressed with the same formalism expressing Properties(P2) and with entities coming from elements in E. E.g. in (13.i) we have the formalism of the Vocabulary Property; in (13.ii), the formalism of the Nucleus Property, and in (13.iii) to (13.viii), the formalism of Arrowing Properties with ';' being the notation of disjonctive arrowing (i.e. (13.iii) expresses: a $PREn$ arrows either to a preceding Nn or to a preceding $PREn$ or to a preceding $ADJn$). The general conjecture is that if General semantic conditions regarding Arrowing pairs are satisfied (see preceding Section 8), and if crossing Arrowing pairs are not allowed (i.e. if there are no two Arrowing pairs such that $Sy_k \rightarrow Sy_i, Sy_l \rightarrow Sy_j$), the graph(s) obtained by (12) are the same as the ones obtained by (13).

The final track till now proceded towards Projections is related, as the previous one, with the description of *ID's* and leads to the description of the whole sentence. It can be exemplified by the following (1) which instantiates the general formulation of (2); Qn_i $(i \geq 0)$ designates a nuclear phrase with a *cat* of the *wh* type - either relative pronouns or a complementizer - or the empty $Qn0$ assumed in the initial position of the string, and Vn_i $(i \geq 1)$ designates a nuclear verbal phrase.

1. $()_{QnO}$ Pierre $(\text{croit})_{Vn1}$ $(\text{que})_{Qn1}$ la fille $(\text{à qui})_{Qn2}$ Marie $(\text{a envoyé})_{Vn3}$ un message $(\text{prétend})_{Vn3}$ $(\text{qu'})_{Qn3}$ elle $(\text{n'ira pas})_{Vn4}$ à la réunion.

2. A Vn closes the first opened Qn to its left.

Following (2), *(croit)*$_{Vn1}$ closes the empty *()*$_{QnO}$ which is assumed at the beginning; *(que)*$_{Qn1}$ and *(à qui)*$_{Qn2}$ are, in this order, the next open *Qn's*; *(a envoyé)*$_{Vn3}$ closes the first *Qn's* to its left, i.e. *(à qui)*$_{Qn2}$, and *(que)*$_{Qn1}$ remains as the only non closed *Qn*. It must be closed by the next *Vn* which follows *(a envoyé)*$_{Vn3}$, i.e. *(prétend)*$_{Vn3}$. In the *(prétend)*$_{Vn3}$ position there is no open *Qn* and thus no *Vn* can be found in the input string before finding a *Qn*: this is *(qu')*$_{Qn3}$ which will be closed by *(n'ira pas)*$_{Vn4}$ (see a less general presentation of this Projection in [Hagège & Bès 99]).

This Projection, once again, borders with a more general Principle because it works - with non-trivial exceptions coming, among other factors, from ellipsis and coordination - with respect to significant sets of structures in French, Portuguese and, very probably, Spanish[17].

The general idea of this last but not least candidate to a Projection/Principle statement is that NL strings can be characterised in terms of detectable entities which, in opening them, function as flags wich indicate the very beginning of a particular kind of strings. They are the correlation of the *K1F* and diabolic transition peculiarites, which, in many cases, allow us to detect immediate transitions between *cat's* pointing to closure positions. In parsing specification, this idea was exploited in a very rudimentary form in [Bès 93]; it is in [Hagège 00] that it was systematically pursued and implemented. [Hagège 00] class *cat's* in terms of *cat's* which either always, never or sometimes can open or close Portuguese nominal nuclear phrases, the information on *cat's* being formally derived from system CAT and P2 (see preceding Section 9.3)[18]. The same idea with the same AF analyser is also exploited in [Rodier 00]; besides those in nuclear phrases and those embedded in *wh* phrases, flags can also be detected in subordinate sentences, at least in French and Spanish (see [Bès 01c].

9.7 5P references

We include in this Section references to published, unpublished and ongoing work related to 5P. GRIL documentation can be consulted at http://lgril.univ-bpclermont.fr.

[Bès & Jurie 89]

Gabriel G. Bès & Pierre-François Jurie. *Métalangage descriptif pour la phrase simple et la phrase enchâssée*. Rapport du projet ESPRIT 393 ACORD, 1989.

[Jurie & [Bès 92]

Pierre-François Jurie & Gabriel G. Bès. "The control of UCG grammars". In *The Construction of a Natural Language and Graphics Interface; Results and Perspectives from the Acord Project*, Springer, 1992, p. 47-64.

[Bès 97a]

Gabriel G. Bès. "L'observation du syntagme nominal noyau en français". Not accepted communication proposed to Paris *CSSP 97*, 1997.

[Bès 97b]

Gabriel G. Bès. "Projection revisited". Not accepted communication proposed to Aix-en-Provence *Formal Grammar 97*, 1997.

[Bès 97c]

Gabriel G. Bès. *P, P & P; P1 ou P de Protocoles, P2 ou P de Propriétés, P3 ou P de Processus*. GRIL, Rapport de recherche, 1997.

[Bès 98a]

Gabriel G. Bès. *P, P & P*. Seminar at Xerox Research Center, Grenoble, November 1998.

[17] Nominal phrases and relatives without relative pronouns are very strong counter examples in English.

[18] We thank Salah Aït-Mokhtar with whom the first named author of this document thorougly discussed about French flags in the 93 summer.

[Bès 98b]

Gabriel G. Bès. *Un sous-ensemble des P2 du français*. GRIL, Rapport de recherche, 1998.

[Bès 99a]

Gabriel G. Bès. "La phrase verbale noyau en français". In *Recherches sur le français parlé*, N° 15, 1999, p. 273-358.

[Bès 99b] Gabriel G. Bès. *Satisfaction et substitution*. GRIL, Rapport de recherche, 1999.

[Bès 99c]

Gabriel G. Bès. *5P - Pouvoir expressif des Propriétés (P2)*. GRIL, Rapport de recherche, 1999.

[Bès 99d]

Gabriel G. Bès. *Le Générateur de Modèles*. GRIL, Rapport de recherche, 1999.

[Bès 99e]

Gabriel G. Bès. *Propriétés et Projections*. GRIL, Rapport de recherche, 1999.

[Bès 99f]

Gabriel G. Bès. *Rapport sur le dossier d'Habilitation à diriger des recherches de Philippe Blache*. Report for Université de Paris VII, December 1999.

[Bès 00]

Gabriel G. Bès. *Vers le parsage de la phrase*. GRIL, Rapport de recherche, 2000.

[Bès 01a]

Gabriel G. Bès. *Categories for 5P*. GRIL, Rapport de recherche, 2001.

[Bès 01b]

Gabriel G. Bès. *Empiricité en linguistique et grammaire de Montague : la sémantique en 5P et la compositionnalité*. GRIL, Rapport de recherche, 2001.

[Bès 01c]

Gabriel G. Bès. *Flags in strings*. GRIL, Rapport de recherche, in preparation.

[Bès & Blache 99]

Gabriel G. Bès & Philippe Blache. "Propriétés et analyse d'un langage". In *TALN*, Cargèse, July 1999, p. 45-54.

[Bès & Coheur 99]

Gabriel G. Bès & Luisa Coheur. *Propriedades e projecçaos*. Rapport de recherche. GRIL & INESC, Clermont-Fd & Lisbon,1999.

[Bès & Coheur 00]

Gabriel G. Bès & Luisa Coheur. *Le Générateur de Modèles*. Rapport de recherche. GRIL & INESC, Clermont-Fd & Lisbon, 2000.

[Bès, Hagège & Coheur 99]

Gabriel G. Bès, Caroline Hagège & Luisa Coheur. "Des Propriétés linguistiques à l'analyse d'une langue". In *VEXTAL*, Venise, Italie, Novembre 1999.

[Hagège 00]

Caroline Hagège. *Analyse syntaxique automatique du portugais.* Thèse; GRIL, Université Blaise-Pascal, 2000.

[Hagège & Bès 98]

Caroline Hagège & Gabriel G. Bès. "Da observaçao de propriedades linguísticas à sua formalização numa gramática do processamento da língua. In *PROPOR' 98*, Porto Alegre, Brasil, November 98.

[Hagège & Bès 99]

Caroline Hagège & Gabriel G. Bès. "Delimitação das construções relativas e completivas na análise de superfície de textos". In *PROPOR' 99*, Evora, Portugal, September 99.

[Rodier 00]

Emmanuelle Rodier. *Diagnostiqueur générique de langues contrôlées.* Thèse; GRIL, Université Blaise-Pascal, 2000.