[Proceedings of the *Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages*, Colgate University, Hamilton, New York, August 14-16, 1985]

STATIC GRAMMARS

A formalism for the description

of linguistic models

Bernard VAUQUOIS

Sylviane CHAPPUY

G.E.T.A Groupe D'Etude pour la Traduction Automatique
I.F.C.I Institut de Formation et de Conseil en Informatique

ABSTRACT

For a linguistic model it is necessary, first of all, to define the mapping between the strings of words of a language and their structural organisation, given that with transducers there are many ways of obtaining the same result using different strategies.

This mapping which we will call a "static grammar'"is independent of the analysis, generation or whatever strategy adopted. Moreover the formalism of a static grammar is not affected by the choice or number of interpretation levels.

Such a grammar is the "reference" for any dynamic modular rule organisation, whether analysis or generation.

We present, here, a "static grammar" formalism recently developed at Grenoble (G.E.T.A. Groupe d'Etude pour la Traduction Automatique), under the supervision of Prof. B. Vauquois.

Using this formalism, any given language can be described as a series of "charts". Each "chart" describes how a certain group of strings corresponds to the structure associated with this group of strings (this structure is a valid and complete substructure of the linguistic model). The structures of all the sentences of a language for a given linguistic model can be described by means of a series of chart inter-references.

The static grammar is used as a base for writing dynamic analysis and generation modules, however, the static grammar does not concern itself with strategic, combinatorial, ambiguity problems or the choice of structures related to dynamic grammars.

We will present here several examples of charts and discuss the dynamic use of these static grammars.

INDEX TERMS

Formal grammars
Transducers
Computer Aided Translation
Static Grammar
Formalism

1. Problem of analysis and generation of natural language in machine translation.

a) The idea of interpretation levels.

Even at the early steps of research in Machine Translation, when V.Yngve proposed a "framework for a syntactic translation", the idea of a "structural descriptor" was the basic concept of second generation M.T.systems.

At the same time, S. Lamb described the stratificational approach that was performing at Berkeley.

This was then followed by the presentation of theoretical papers by I.Melchouk and A.Jokolwski showing the development of a "meaning-text" model through a sequence of intermediate levels of interpretation.

At the same time, the first M.T.system realized at C.E.T.A. (from Russian to French) was organized in a sequence of levels (morphological interpretation, bracketting of a sentence according to morpho-syntactic classes, syntactic relations between the words of a sentence, logical relations, etc...). In fact, this notion of level of interpretation is the only one which can give a formal representation of "meaning". Indeed, a level of interpretation has to be described as a reference universe with atomic primitives and rules for the significance of valid formulae.

To speak about meaning without any such reference is "meaningless".

So, we will speak about levels of interpretation, defining each level by its primitives, its data structure and its rules of building formulae.

At the text level (considered as a string of characters), which is called the level 0 of interpretation, two sentences (ie: two strings) are different (so are interpreted differently) if they are not the same string of symbols.

For any level, two formulae are different if they don't match exactly on the agreed data structure.

Two or more sentences may be represented by the same formula for a suitable level of interpretation. In such a case, these sentences are considered as equivalent in the universe representing that level.

By defining levels more and more independent of the surface, we hope to collect the maximum number of "synonymous" sentences in a single formula in the analysis phase, and, for such a formula, we hope to be able to build the maximum number of paraphrases.

A level j is said more powerful (or further from the surface) than a level k if the paraphrasing possibility is greater from level j than from level k.

The preceding notions are extremely useful for M.T.system where the syntactic structures of the source and target languages do not necessarily correspond. Consequently, it will be interesting to reach levels of interpretation which do not depend any longer morphological and syntactical constraints of any languages.

b) Examples of levels of interpretation and structures of representation.

1. First of all, it is necessary to substitute a collection of linguistic information for each word or each idiomatic expression considered as a string of characters. That is the morphological level. The linguistic information may vary according to the requirement imposed by the higher levels which are expected in the model. At least, it is necessary to know: the lexical unit, the grammatical attributes associated to the form of the word and some syntactic and semantic properties. At this level of interpretation, it may happen that two different strings, as word forms, have the same interpretation, like:

FARTHER and FURTHER.

But, in any case one single string has several interpretation, like:

LOCK

(noun with different meanings, ie. with different semantic features or verb with two different syntactic properties and several semantic features).

2. Then, it may be interesting to consider an interpretation of every sentence in terms of phrase bracketing. Such a level must be considered as the most superficial one. Let us call it the level of "syntactic class".

3. Then, a higher level of interpretation, which, consequently has a higher power of paraphrasing, is the level of syntactic dependencies. Let us call it, the level of "syntactic functions", this level remains close to the syntactic constraints of the language.

4. In order to escape from some of the constraints, it has been found interesting to introduce a level of 'logical relations'. By considering that in every language some words are predicative (ie. could be compared to predicates in logic), it is possible to exhibit the "logical structure" of a sentence in terms of predicates and their arguments.

Certainly, this kind of relations does not suffice for a complete coverage of any sentence, indeed, all circumstantial connections do not belong to this predicate-argument relationship, it is convenient to introduce the level of 'semantic relations".

The logical relation implicitly implies semantic relation, depending on the nature of the predicate, the place of the argument and the semantic features of the argument; however, in most cases, it is not necessary

to compute explicitly the value of such a semantic relation.

This level giving the logical relations complemented by the semantic relations seems to fulfil the conditions expected for a suitable "structural descriptor".

For each of the these levels, the chosen structure is a tree where each node is decorated by a set of labels.

At the PS level (Phrase Structure, ie. bracketting) it is understood implicitly that any arc from a node B to its mother A means "B is a constituent of A".

At the SF (Syntactic Functions) and LSR (Logical and Semantic Relations), some labels on node B indicate a relation from node B to the mother node A.

Consequently, the tree structure of PS level cannot match the tree structure of SF and LSR levels.

Let us, now, give an illustration by the following example:

(1) beaucoup d'équations sont résolues par itération.
(2) on résoud beaucoup d'équations par itération.
(3) beaucoup d'équations se résolvent par itération.
(4) il se résoud beaucoup d'équations par itération.

These four sentences have the same translation in English,

(5) many equations are solved by iteration.

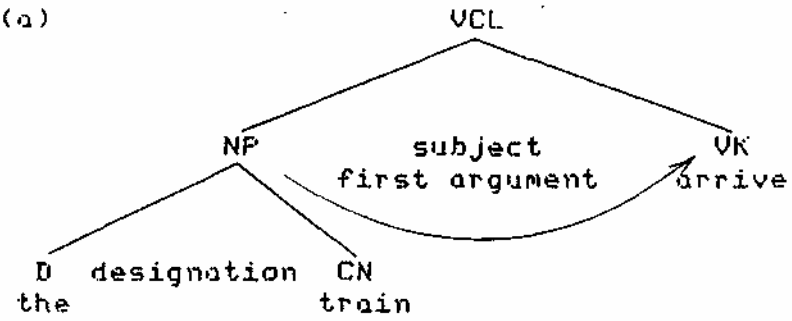
Only (1) to (5) is "word-word", to obtain the same translation for each of the four French sentences, it is necessary to give the same interpretation for the four sentences.

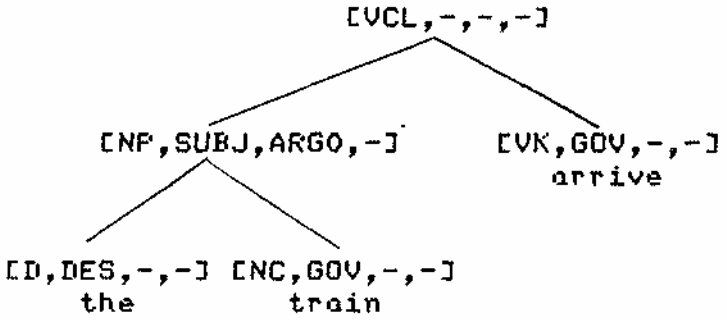
For the analysis phase, first, it seems natural to proceed sequentially, however, at the lower level (especially at the PS level), many structural ambiguities cannot be solved; consequently, they must be conveyed to the following level and so on. Moreover, every time that the analysis fails at the top level, this sentence is lost even if the lower levels are successfully obtained. Finally, there is not possible interaction between the different levels. The ideal situation would be a parallel processing, but, as we have seen, the tree structures associated with the different levels can not compared. In 1974, the Grenoble Group decided that the solution was the definition of a unique data structure acceptable at all levels.

A decorated tree structure in which the geometry itself is connected with the weakest linguistic interpretation seems to be an adequate data structure. The shape of the tree represents generally the bracketting of a sentence into phrases. The labels dealing with node properties (for example, the value of any lexical attributes) belong to the decoration of this node. The labels dealing with relationships between two nodes belong to the decoration of the source node, the target node is implicitly addressed by the name of the label.

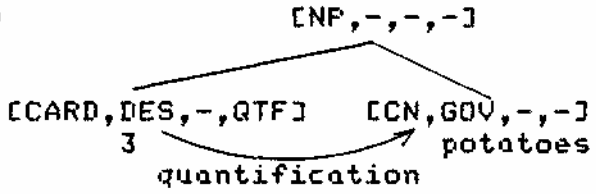
Example:

(a)

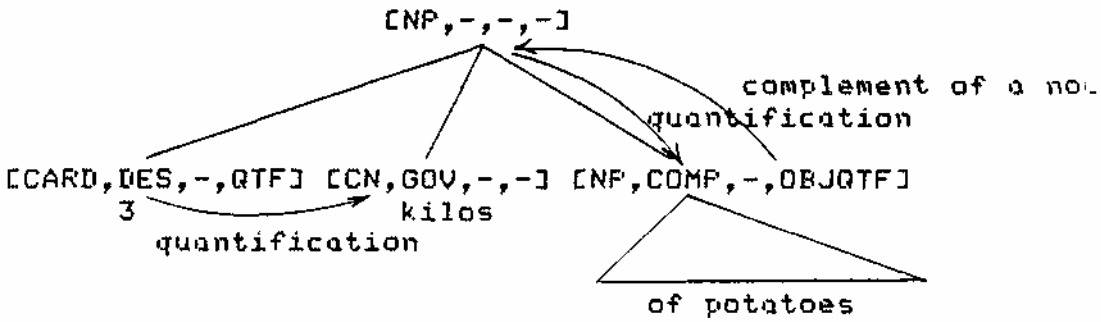

VCL
NP
subject
first argument
VK
arrive
D
the
designation
CN
train


[VCL,-,-,-]
[NP,SUBJ,ARGO,-]
[VK,GOV,-,-]
arrive
[D,DES,-,-]
the
[NC,GOV,-,-]
train

(b)

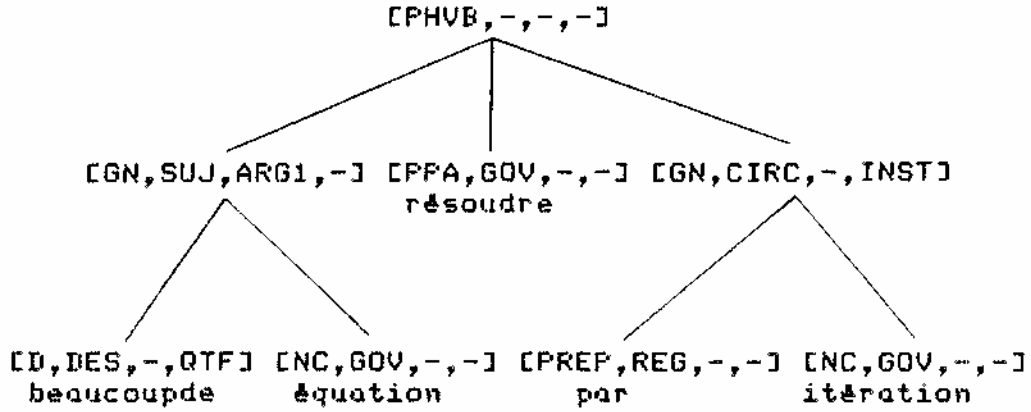

[NP,-,-,-]
[CARD,DES,-,QTF]
3
[CN,GOV,-,-]
potatoes
quantification

(c)

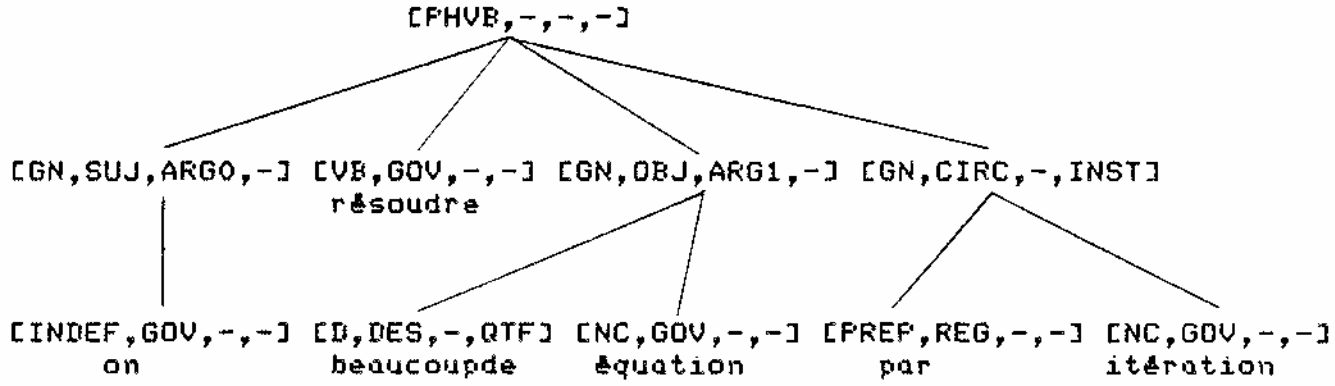

[NP,-,-,-]
complement of a no
quantification
[CARD,DES,-,QTF]
3
[CN,GOV,-,-]
kilos
[NP,COMP,-,OBJQTF]
quantification
of potatoes

Let us look at the multi-level structures associated with the four French sentences:

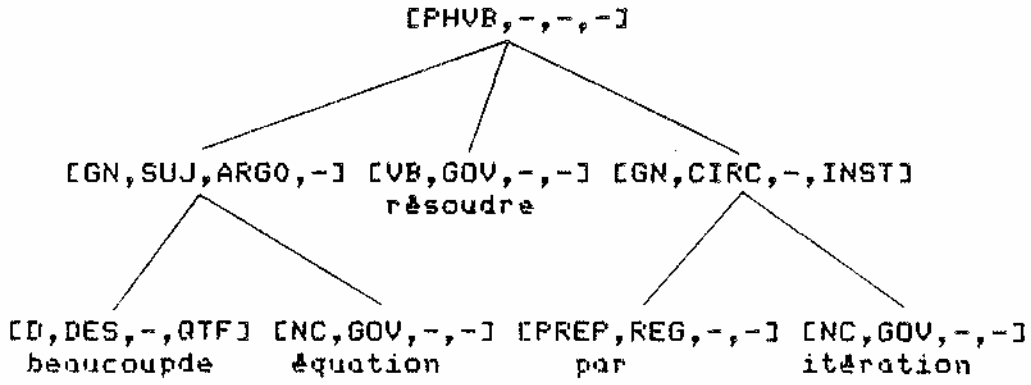
(1) beaucoup d'équations sont résolues par itération



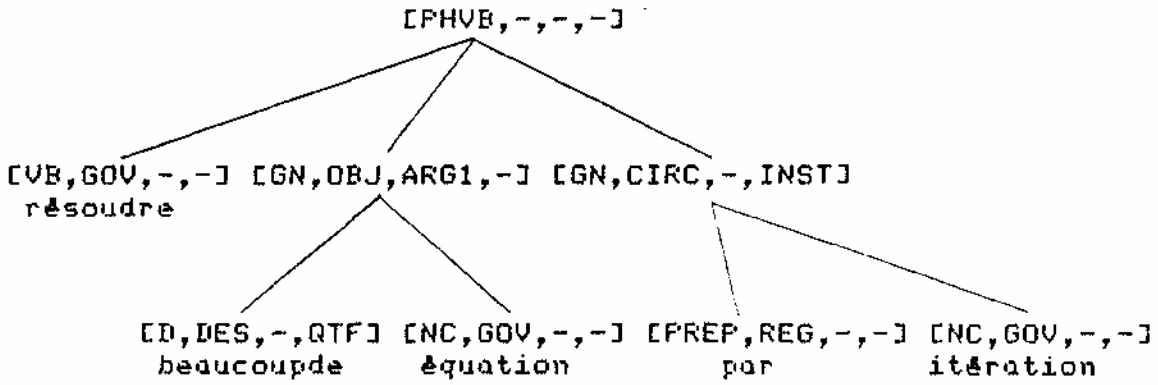
(2) on résoud beaucoup d'équation par itération



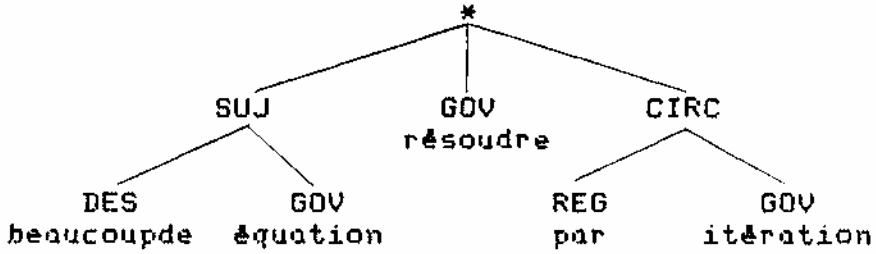
(3) beaucoup d'équations se résolvent par itération



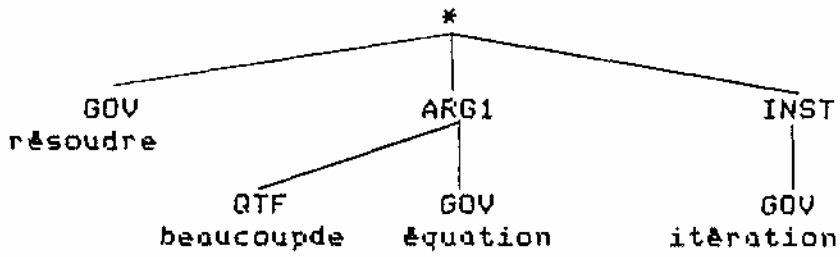
(4) il se résoud beaucoup d'équations par itération



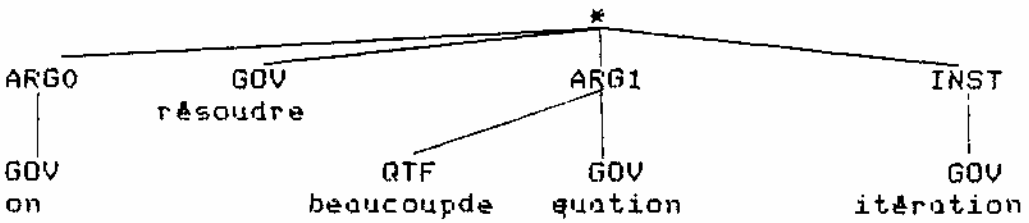
The paraphrases (1) and (3) have the same SF-structure,



To recognize (4) as a "paraphrase", we must consider the RLS-structu-



The RLS-structure for (1) is:



We must define a structural equivalence to recognize (2) as paraphrase because of the indefined value of the representative "on" French.

The different levels can interact in both directions; a failure at some level can be repaired by a grammar belonging to another one; in case of impossibility of some sentences to achieve the highest level of interpretation, the less ambitious levels are not lost and they can deliver, nevertheless, a satisfactory translation; this approach enables the system to manage "fail-soft" techniques which are appreciated in practical machine translations.

c) Inadequacy of Generative grammars.

The existence of parsers does not solve all that is needed as software tools for a translation process. Even if dictionary look-up algorithms are included in the whole system, the structural transfer cannot be handled. Furthermore, if the intermediate structure is expected at a deeper level of interpretation, some other tool is necessary to transform the results of the surface analysis. Also, it may be requested that some deep analysis could be obtained without parsing through the surface structure; indeed, sequential levels of interpretation have the great disadvantage of carrying a lot of ambiguities which cannot be solved at the early stages.

Second, the obtained structure is a by-product which is not always compatible with a consistent linguistic interpretation.

Third, such grammars are defined by the total set of rules. If the grammar is not restricted to the definition of a small sub-language and is expected to cover large quantities of various texts in a given natural language, then the number of rules becomes extremely important. Under such conditions, debugging a grammar containing hundreds of rules is an impossible task. Every time a rule is modified in order to accept new sentences, this modification may prevent the recognition of former sentences which were accepted.

Fourth, in case of failure, for a given sentence, an empty result is obtained for that sentence. It is extremely unpleasant to block the translation process for such sentences because the formal grammar is not totally adequate or because the sentence itself is not well-formed. A translation should be obtained even for badly written input texts.

Fifth, such grammars can be considered as static sets of rules. The grammar, by itself, has no effect on how to proceed with rules. Consequently, the algorithm of the parser is a combinatorial one, in order to detect all possible structures. The only possible restriction against the combinatorial explosion comes from the algorithm itself; in any case, the linguist cannot indicate through the grammar rules how to speed up the process. Another consequence of the combinatorial effect

is the creation of parasite ambiguities.

The desired structural descriptor must be defined "a priori" by the mapping between strings and structures (and must not be a by-product of some derivational grammar). Consequently, the dynamic process of analysis or generation can be segmented into small modules which transformed the data structure. With such an organisation, the generative grammars and their associated processes are replaced by "transduction grammars" and "transducer automata".

Transducer automata carry out transformations on the input structure until the output structure, whose interpretation level is predefined is obtained. Such automata are most frequently used for the translation of natural languages.

MIND - system (1970) ; M. Kay and R. Kaplan, Rand.
ATN (1970) ; W. Woods, BBN.
Q-Systems (1970) ; A. Colmerauer, Montreal.
ATEF (1972) and CETA (1974) ; J. Chauché, Grenoble.
PLATON (1974) ; M. Nagao and J. Tsujii, Kyoto.
REZO (1975) ; Stewart.
ROBRA (1977) and TRANSF (1978) ; M. Quezel-Amtarunaz and P. Guillaume and C. Boitet.
SYGMOR (1978) ; D. Jaeger, Grenoble.
GRADE (1983) ; Nagao, Tsujii, Nakamura.

Transduction grammars are programmed directly for analysis or generation. The output of the analysis is defined "a priori", the transducer may be used in many different ways according to some chosen strategy ; modular architecture offers a great variety of heuristic processes and, by means of the control on this architecture, it is easy to perform an analysis, guided for a large part by the text and not uniquely by the grammar. For generation, the transducer is used but clipped for the choice of the output syntactic structures (for some texts, a translation with a syntax remaining as close as possible to the source system may be demanded, whereas for other texts, some preassigned style may be preferred). Transduction grammar refers only to itself.

For a linguistic model it is necessary first of all to define the mapping between the strings of words of a language and their structural organisation given that with transducers there are many ways of obtaining the same results using different strategies. This mapping which we will call a "Static Grammar" is independent of the analysis, generation or whatever strategy adopted. Moreover the formalism of a static grammar is not affected by the choice or number of interpretation levels (as well as the data structure for these levels remains a decorated tree). Such a grammar is "the reference" for any dynamic modular rule organisation.

2. Elements for a formalized description of a static grammar.

a) Problem of mapping the strings and the structures according to the levels of interpretation.

The structural static grammar(whose formalism is presented here) establishes a correspondence between strings of words in a given language and their appropriate tree structure at various levels of interpretation.

* Various constraints have been imposed on the formalism.

- The notion of "charts" instead of that of rules. A chart is a mapping between valid sub-strings of the language and their corresponding structures.

- The formalism must be flexible enough to allow choice in writing the charts; the linguist can make a description of a linguistic model using few but complex charts or using a great number of simple charts.

- The formalism must allow for the static grammar to be easily understood, up-dated and enlarged.

* Various constraints have been imposed on the linguist.

- The necessary declaration of all attributes with their types and values, the only information available to the model.

Examples:

CAT=V,N,A,D,R,S,....
Syntactic category of: verb, noun, adjunct, determiner, representative,...

SUBA=ADJ,ADV,CARD,ORD,ADJM,..„
Sub-category of adjunct: noun adjunct, adverb, cardinal number,...

NUM=SIN,PLU.
Number= singular, plural.

SR=MANNER,ACCOMP,ANALOG,CONCESS,QFIER,OBJQF,.. .
Semantic relation= manner, accompaniment, analogous, concession, quantifier, quantified object,...

K=AP,NP,ADVP,NUMP,VCL,RELCL,SCL,PARTCL,INFCL,VK.
Morpho-syntactic classes: adjectival phrase, noun phrase, adverbial phrase, numerical phrase, verb clause, relative clause, subordinate clause, infinitive clause, verbal kernel.

etc...

-The necessary organisation of a hierarchy of classes and some ordering between the charts which describe the "local event".

Example:

Let us consider that:

-the description of NUMP is independent of the description of the other morpho-syntactic classes, in the hierarchy the NUMP level is 0;

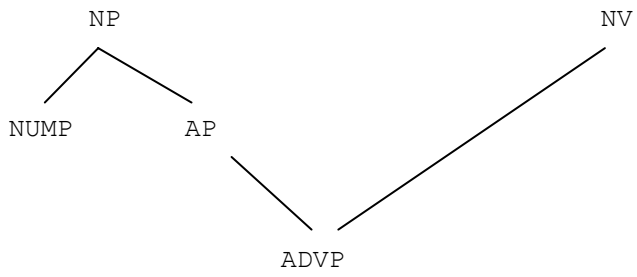
-the description of ADVP is independent of the other description;

-the description of AP is dependant of the description of NUMP but independent of the other description;

-the description of a NP is dependant of the description of AP (and consequently of ADVP) and of NUMP but is independent of the description of the other classes;

-etc . ..

We can establish the following hierarchy:



It is a hierarchy on the simple phrase, but for the complex phrase and for the clauses, the description of a phrase or a clause can be dependent on another phrase or clause of higher hierarchy.

b). Description of the formalism.

The notion of "chart".

The chart is a partial mapping between a set of valid substrings of the language and their structures, the formalism makes it possible to have many strings (x) on a simple chart.

The chart is divided into three zones :

1. the first zone shows the correspondence between a tree structure and a sequence of words in sequence.

2. the second zone indicates the constraints on the corresponding sequence and structure described in zone 1 depending on the presence and the decoration of the elements of the string.

3. the third zone indicates the relations between the decorations of the different nodes of the structure and the decorations of the different nodes of the string.

b.1. Development of ZONE 1 :

This zone shows the correspondence between a family of tree structures and a family of strings of a language.

The string

The formalism allows for each chart to cover a family of strings thus limiting the number of charts necessary and therefore reducing the size and complexity of a static grammar. This family of strings is described by a sequence of elements, which are either optional, obligatory, iterative.

Because of the optional elements, an intrinsic constraint is imposed for the formalism, that of maximum coverage. The string described by the chart is the largest possible string corresponding to one of the strings of the family described by the chart.

An element "X" of a string may be :

- obligatory : we note X
- optional : we note (X)
- obligatory and iterative : we note X+
- optional and iterative : we note X*

An element of a string may be any of the following :

- a key-word, that is one referenced directly by its lexical unit (eg. "not", ".", "that", "do"),
- a syntactic class of words possibly determined by a subclass

N/CN noun/common noun
V/INF verb/infinitive
V/-PAP verb/except for participle
V/INF PAP verb/infinitive or past participle
etc. . .

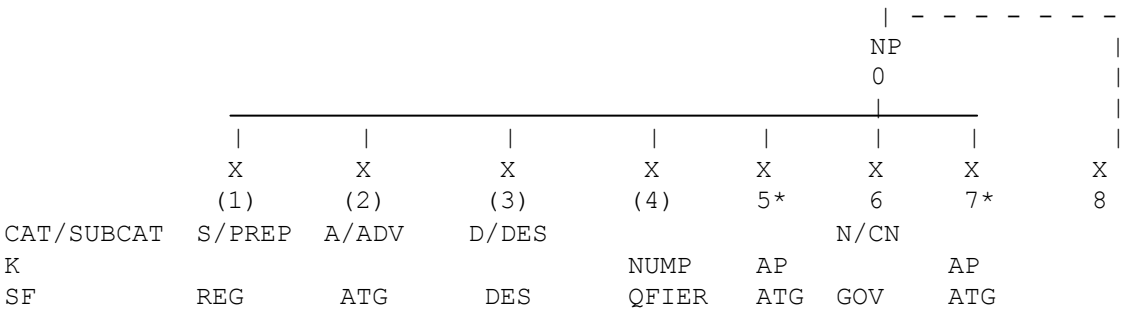
- a syntagmatic group

NP - noun phrase
INFCL - infinitive clause

In such a way, a substring of the string described in the chart is represented. A structure has already been associated with this substring in another chart (which must be referenced) and the description in this chart used this sub-description only when this sub-description can be totally integrated (independently of the context of the referenced chart) eg. NP as SUJ/ARGO in a verbal clause or completed (independently of the context of the referenced chart) by another group already described, by elements of strings referenced (by their syntactic categories or their Lexical Unit, etc).

The structure:

Given the following string and associated structure:



(Element 8 is the context which will be discussed later, it validates the phrase described as NP in the model).

Elements 1, 2, 3, 4, 5, 7 are optional, 5 and 7 iterative, element 6 is obligatory.

1, 7, 3, 6 are referred to by their syntactic category.
4,5 and 7 reference already described sub-strings and the associated sub-structures, which have been integrated here into the noun phrase being described.

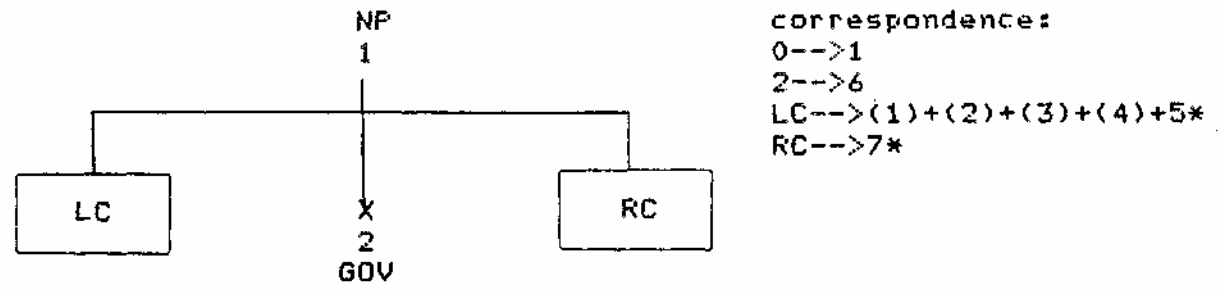
Once a phrase has been described in a given chart, it can be referred to in a new chart in several ways :

- if no information is required from the elements of the substring, if the description of this sub-string exist in an other chart and if this description is valid in this new chart, this

description can be referred and be absorbed by the new description using the corresponding syntagmatic class value (K). This is the case with nodes 4, 5 and 7 of the example above.

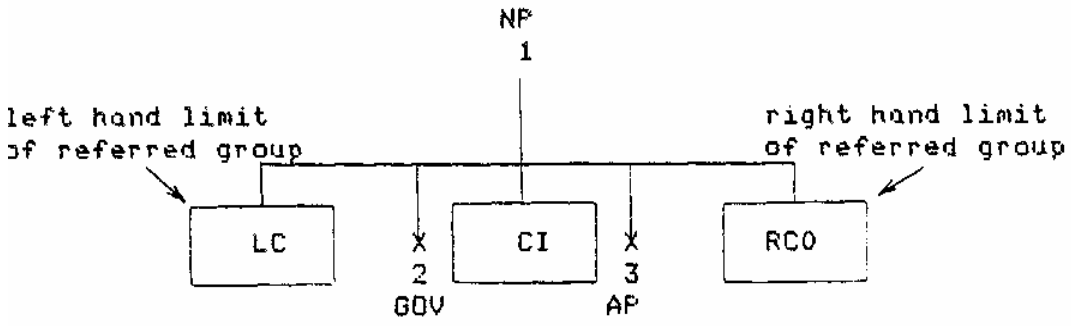
- if information is required from one of the elements of the substring or if this substring has to be modified or completed, or if the structure associated with this substring is different from the structure associated with the substring of the chart referred, the reference will have to be represented in order to indicate one or more of the elements of the substring referred in order that the limits of the substring referred are distinguished from the described string.

A phrase like the above noun phrase could be referred to in the following way:



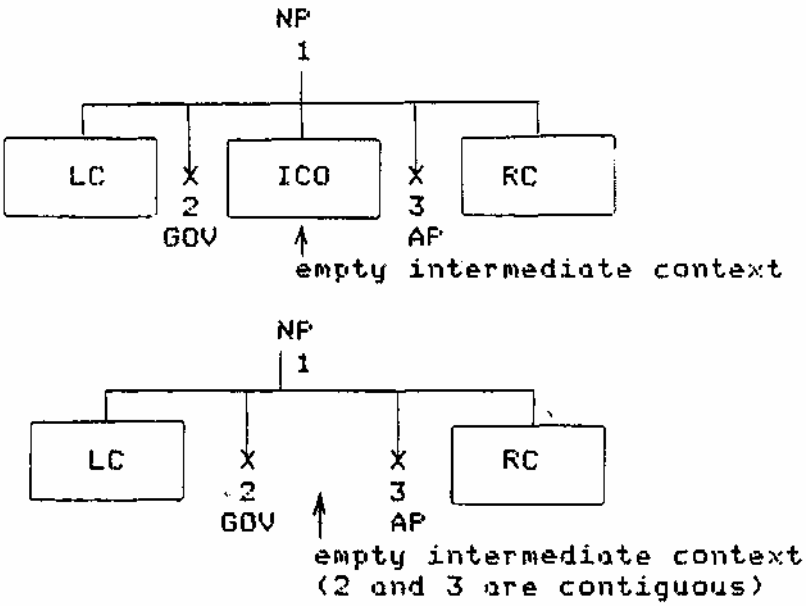
This sub-structure is a complete structure, everything to the left of the governor in the referred noun phrase is contained in the lefthand context LC, likewise everything to the right of the governor is in the righthand context RC. The lefthand side context LC determines the limit to the left of the phrase referred, the righthand side context RC determines the limit to the right. In this case, the reference can have access to the governor of the referred noun phrase and hence to its attributes, any other node could be emphasised by checking particular values of its attributes.

The notion of intermediate context IC allows several particular elements within the phrase referred to be emphasised.

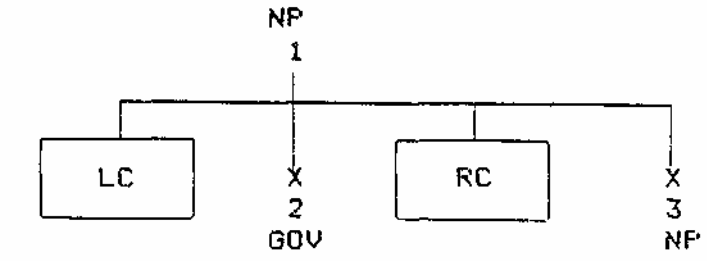


A lefthand context (righthand respectively) which is required to be empty would be represented LC0 <RC0 respectively). In the above example the highlighted AP is the last element to the right of the group referred.

An empty intermediate context can be represented as IC0 or can be omitted from the structure, as since the structure is totally described, the juxtaposition of the two elements implies their contiguity since everything - which is part of the referred group (between LC and RC) has to be expressed, likewise everything which is expressed between LC and RC must be part of the referred group.



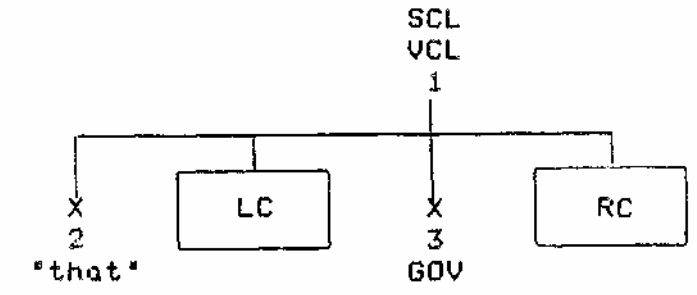
If when the phrase is referenced, the correspondence between sub-string and sub-structure in the referenced chart and in the new chart is the same, with same root same string same structure, there will be no problem in situating it. On looking at the new chart we will immediately know what type of substructure, what type of substrings the structure and of the string described) are grouped together by the reference



This diagram describes the structure of a noun phrase using already made descriptions of the syntagmatic phrases 1 and 3, this diagram in fact makes explicit the relationship existing between strings of elements of phrase 1 and phrase 3, which is completely absorbed into the new description. The referred phrase 1 which was valid in a certain context in the chart which described it, is referred to here out of context to be completed and to form a new valid noun phrase in a new context which will likewise have to be made explicit.

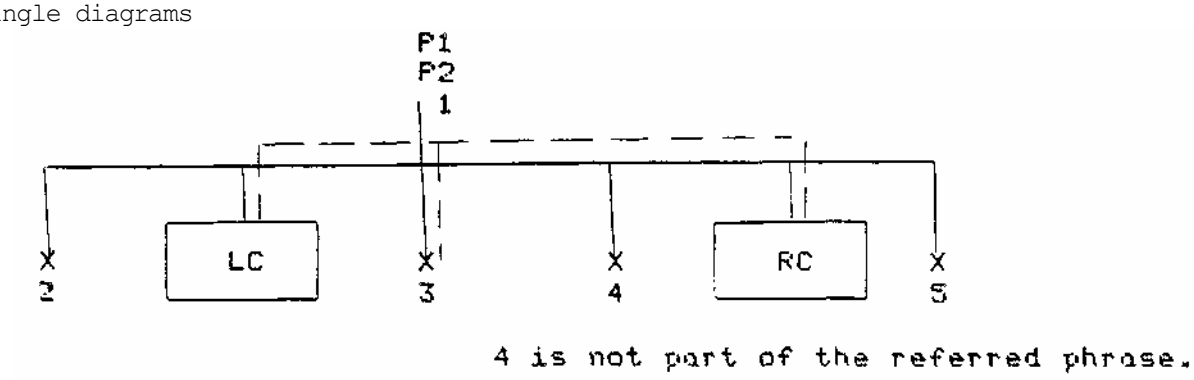
The string corresponding to this structure is the string of elements of the referred phrase 1 concatenated with the string of elements of the referred phrase 3. The corresponding structure is the structure of the referred chart 1 (corresponding to sub-string LC+2+RC) absorbing to the right the structure of the referred phrase 3. The string described in this chart is LC+2+RC+3.

If when referenced two roots are grouped together to form one or the identity of the root is changed, both roots will be given, the absorber and the absorbed, and this fusion will be indicated by a brace



(a SCL as described here is based on the description of a VCL: the string of a SCL is the same as the string of a VCL, but the first element is a subordinator and the name of the syntagmatic class is different, we use the description of the sub-string LC+3+RC which is the description of a VCL to describe the string S("that")+LC+3+RC which is the description of a SCL).

Moreover, the formalism for representing the referred structures must allow for the structure of the group referred to be modified by a different division of the daughter of the root of the structure. As it is necessary to preserve the structure of the referred group (thus allowing identification of the reference) and to recognise the new structure, our formalism has to be able to express both structures in a single diagrams

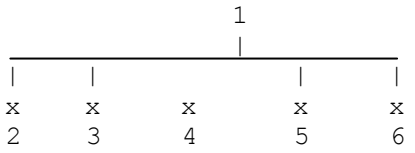


We hope to have covered with this formalism all reference representation problems encountered by linguists.

The relationship string-structure:

By associating a structure to a string certain problems can arise when the elements of the string and associated structure do not correspond.

If an element is present in the string and absent in the structure its presence is indicated in the string but it will not be joined to the structure



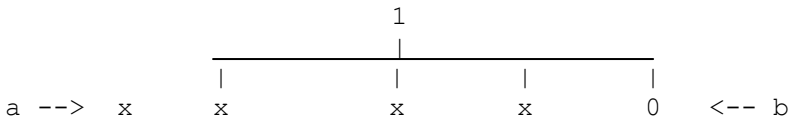
4 is present in the string but not in the structure

(the case arises frequently and enables among others to eliminate keywords which are variabilised : negation, auxiliaries...).

- if the element is present in the structure and absent in the string, its absence in the string will be marked by a "0" this element marked "0" will be attached to the structure (this makes it possible to treat problems like elision).

4 is present in the structure but absent in the string

- if there is a break, between the string and the structure: when the order of the elements of the string is different from the order of the elements of a structure, the formalism must keep a trace of each of there places and a trace of the relationship between these places



element 2 is in a in the string and b in the structure

The context :

It is necessary that the formalism allows for a certain number of elements of the context to the left and to the right on the family of the string described in the chart, to be consulted to valid the correspondence string-structure and before deduce from this correspondence certain attribute values.

The possibility of testing the context limits the range of a chart to only the string corresponding to valid structures in the model.

For example, the static grammar should not describe the string below as corresponding to a valid structure : "the birthday cake" if the right hand context of the string is the following "candles". But for a string such as "the birthday cake candles" the string/structures correspondences validated by the grammar are the following :

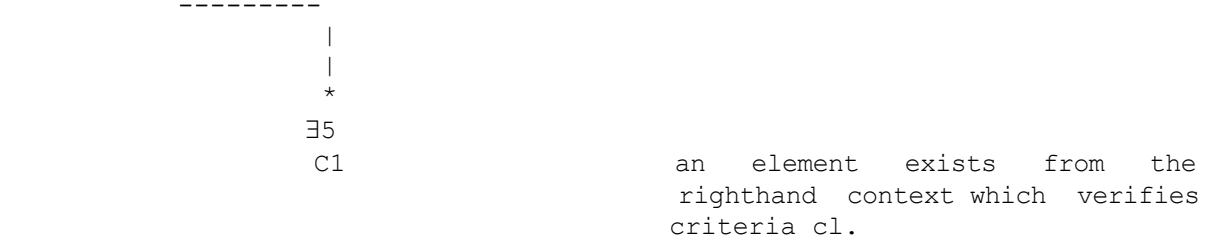
"candles" / NP
"cake candles" / NP
"the birthday cake candles" / NP

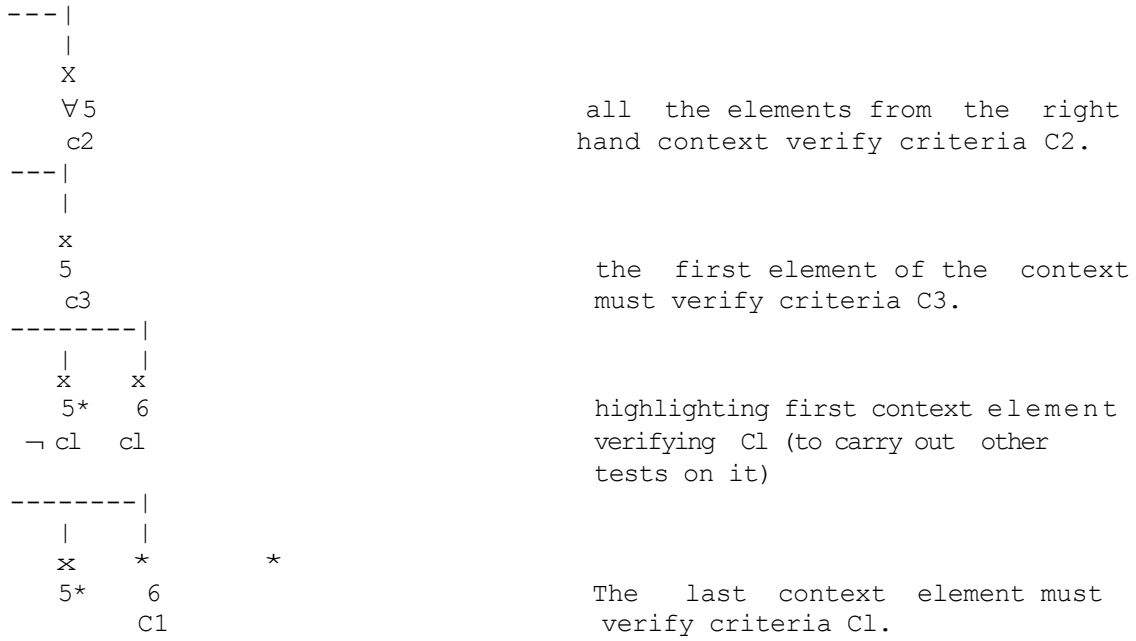
So that the static grammar can use, in a chart describing a complex correspondence (family of strings/structure), one (or more) simpler charts describing the correspondences : family of strings/structure, the references are made independently of the validity context expressed in the referred chart(s), a new validity context being expressed on the new chart.

For this purpose, the context is considered as a list of elements of string and the context elements are treated as elements taken in this list of elements.

To emphasise one (or more) particular elements of the string (for checking a condition) in the list of available context elements, predefined operators are associated with this list "+" and "*" and the universal and existential quantifiers " ∀ " and " ∃ " , with their normal meaning. They represent a condition on the list of context elements taken as a whole. The formalism imposes that the elements (of the list of context elements) mentioned are contiguous (from the right for the lefthand context, from the left for the righthand context). In such a way a particular element can be tested against its predecessors or its successors. If need be the end of context can be expressed by "*".

Several examples are given here of the righthand context.





In this way a particular context element can be situated exactly by multiplying the conditions on its neighbours.

b.2. Development of ZONE 2

This zone expresses the validity constraints of the string-structure correspondence developed in ZONE 1 depending on the existence or not and on the decoration of string elements and of context elements. The role of this zone is to define the family of strings treated by the chart.

(Remark: in order to aid reading the chart, certain constraints have been expressed in zone 1, it is however preferable now to express them in zone 2).

The formalism must be capable of expressing two types of constraints, the constraints concerning the node and the internodal constraints. The constraints concern the existence and the attributes of the elements of the string and they can depend on other constraints of the same nature on one or more other context elements or elements of the string.

. Existence or non existence constraints:

The formalism allows for the presence or absence of an element of the string to be tested, given the element 7,

the absence of 7 is noted: - 7
the presence of 7 is noted: 7

.The operators defined for writing the constraints are the following:

negation	¬
equality	=
inequality	≠
logical and	∨
logical or	^
implication	⊃>
equivalence	<=>

contained (and inverse relation) < and >
no contained < and >

intersection	∩
union	∪

The formalism allows the use of predicates witch will be defined in the appendix of the static grammar. A predicate call is preceded by "$" in order to avoid any ambiguity.

Every non-verification of a constraints in zone 2 entails the non-validity of the string for the string-structure correspondence expressed by the chart.

The fact that certain elements are optional and thus likely to be absent causes problems in understanding the logic of the tests concerning these elements. If CAT(X)=N where X is optional is a condition which is only of interest to us when X is present and must be" written X ⊃ CAT(X)=N, likewise if Y ⊃ CAT(X)=N where X is optional, is a condition which is only of interest to us if X is present and must be written Y ⊃ (X ∧ CAT<X)=N).

The reasons of simplifying the writing we say that any test on an optional element of a string X is implicitly understood as

"presence of X ⊃ < test on X >"

thus, if X optional read CAT(X)=N as: X ⊃ CAT(X)=N

and, Y ⊃ CAT(X)=N as: Y ⊃ (X CAT(X)=N) ⊃ true if X absent.

and, Y ⊃ (CAT(Y)=N ∧ CAT(X)=N) as: Y ⊃ (CAT(Y)=N ∧ (X ⊃ CAT(X)=N)) true if X absent.

In the opposite case, write:

Y ⊃ X ∧ CAT(X)=N and

Y ⊃ (CAT(Y)=N ∧ X ∧ CAT(X)=N) false if X is absent.

Having said this, the logical operators previously mentioned preserve their habitual logical properties and the corresponding truth tables are still valid, bearing in mind that x has the value true if x is present, and false if x is absent.

b.3. Development of ZONE 3

The third zone expresses the relationships which exist between the decoration of different nodes of the structure and the decoration of different nodes of the string.

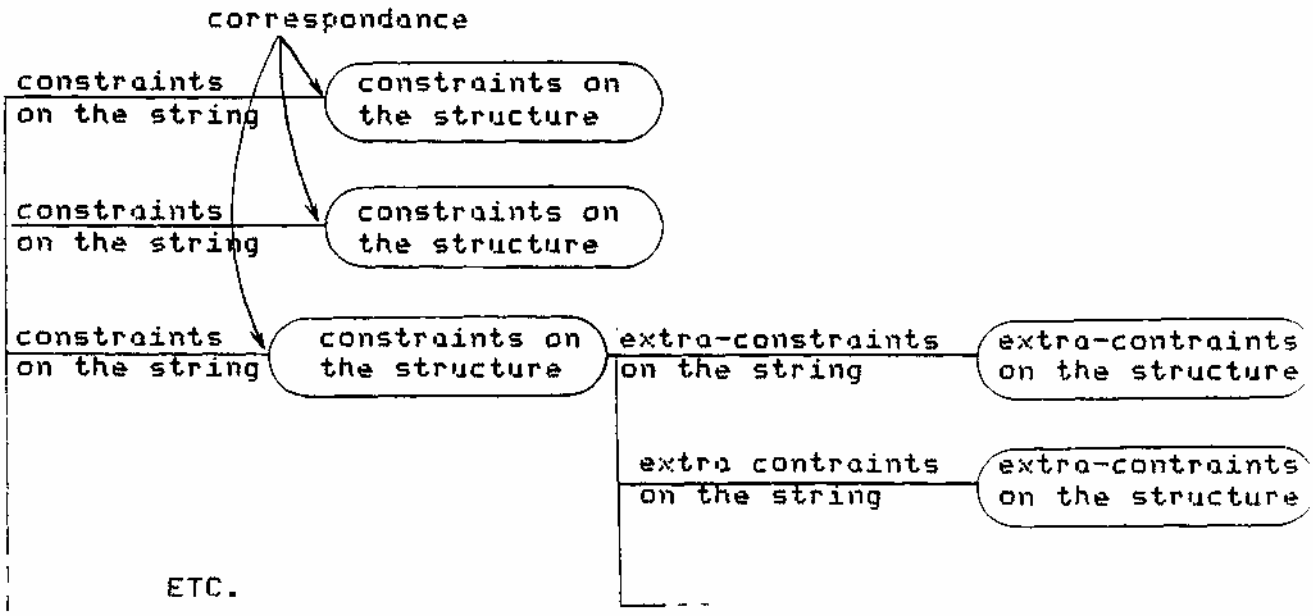
Expression of constraints operators and notion of predicates expressed in zone 2 remain the same in zone 3.

There are two types of constraints and then two sub-zones in this third zone:

In the first, decoration on the elements as elements of the structure independent of the decoration of elements as elements of the string. The correspondences are directly expressed.
ex : $EGAL.CAT(0, g) the category of node 0 and g are equal.

In the second one, decoration on the elements of the structure and decoration on the elements of the string dependent one another.

The decoration correspondences are expressed on a graph of arcs and boxes where the arcs carry the decoration of elements as elements of the string and boxes carrying the decoration of elements as elements of the structure.



Each line of such a graph establishes a correspondence between the decoration of elements of the structure and that of elements of the string.

The lines of such a graph are not compatible amongst themselves. Zone 3

can contain several such graphs.

2.4. OTHER ZONES

Added to these three zones are:

- a zone for examples and remarks. An example of each type of string described by the chart should be given. Eventually remarks on linguistic phenomena which have not been treated in the model, etc., can be given here;

- a zone for the chart heading, indicating the chart number in the grammar, the type of syntagmatic group described, the case treated, the charts referred and the chart which reference the chart in question, etc.;

- a management zone, indicating the author, corrections made, creation date and use in dynamic models.

3. Proposal methodology for static grammar writing.

The formalism which has just been presented is currently used for writing static grammars for English, French, Arabic, Malay. Some of the above mentioned are in the process of being written, some have already been used to guide the writing of grammars:

- structural analysis of French
- syntactic generation of English
- syntactic generation of Arabic (small-scale English-Arabic model).

Several remarks can be made on the methodology of Static Grammar writing.

3.1. A necessary prerequisite for establishing a static grammar is the definition of a complete set of variables (attributes) to describe the given language according to the levels of interpretation adopted and the complexity of the model.

3.2. It is necessary to define a hierarchy among the selected syntagmatic classes of the set of attributes which corresponds to the "natural hierarchy" of simple syntagmatic groups in the language.

3.3. According to the hierarchy defined, the structures are classed according to the given syntagmatic class by:

- elementary
- simple
- complex.

An elementary syntagmatic phrase consist of the most elementary elements: the terminal elements of a given syntagmatic class.

Example: "common noun"+"proper noun", an elementary NP group will be associated with this structure, this strings behave as a "noun" within the NP.

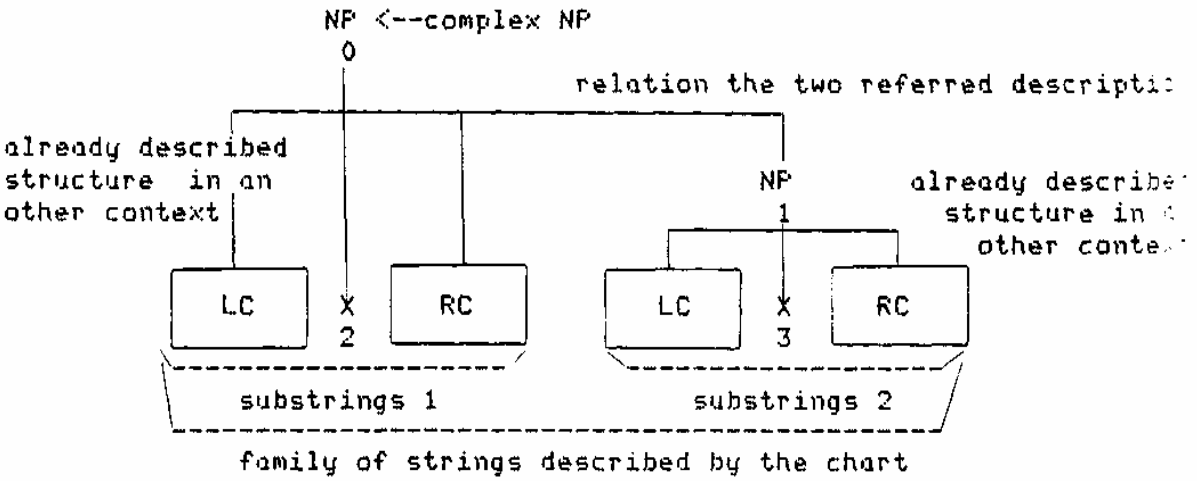
A simple group is a group made up of terminal elements whose syntagmatic class is lower in the hierarchy. Thus a simple AP could contain an ADVP, a simple NP APs or NUMP but a simple AP could not contain a NP or a AP, etc.

A complex group is a group dominating groups of either equal or higher classes.

Examples: AP dominates NP, ADVP dominates NP, NP dominates NP or INFCL *or* PARTCL or RELCL.

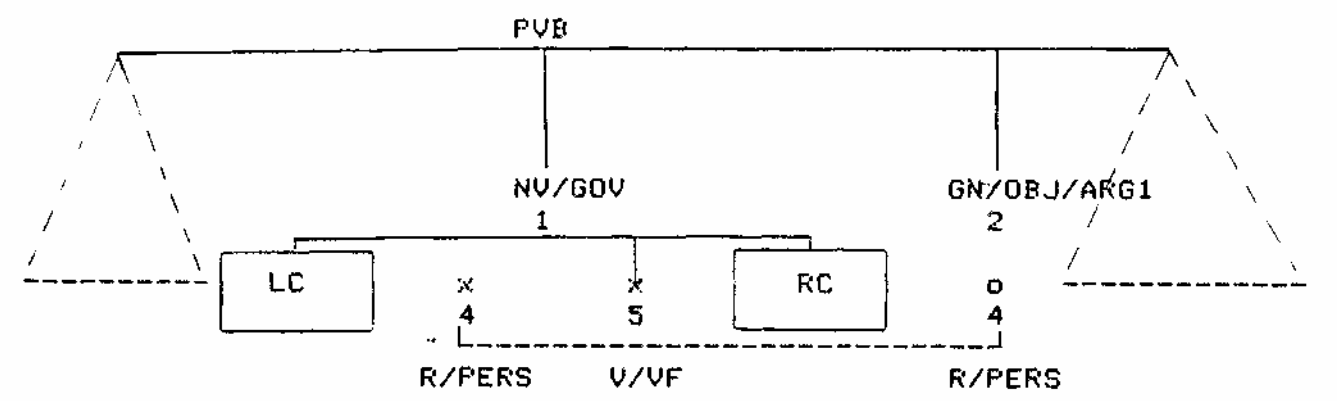
3.4. Remarks on use of references.

a. The charts describing complex phrases most often describe a string-structure correspondence using several charts describing sub string-structure correspondences, in which case it expresses the relationship between the two structures referred, where the string is no longer each of 2 strings but the concatenation of those 2 strings out of reference context but within a new context which has to be expressed.



b. The formalism allows for "combining" of the structures referred in the description. In this way, elements which can be found in the string in the middle of elements corresponding to a syntagmatic group without belonging to this group, do not belong to the description of this group, but can be "combed" within this group when this group is referred (because they are at this place in the string), and placed in an other place in the corresponding structure (because they are not at the same place in the structure).

In this way, personal pronouns, which in French can be found between the elements of the verbal kernel in the string (just before the verb or just before an auxiliary in a compound tense), are not considered in the description of verbal kernel, as they do not belong to it, but they will be combed within the referred verbal kernel (as they are here in the string) when describing clauses which can have pronouns as objects. In the structure, the NP is a verbal kernel "brother".



Example:
'le président leur parlera à 20 heures'

4.Conclusion.

The importance of static grammar is 'threefold':

4.1. The static grammar formalism is a language for specifying linguistic models. A static grammar constitutes the description of linguistic model and serves as a reference for all analysis or generation programs aspiring to calculate this model.

4.2. From any one static grammar, different analysis or generation programs (dynamic grammars) may be written whose strategy and heuristics vary, the equivalence of these programs is assured as each reference the same static grammar.

4.3. Once a program (dynamic grammar) has attained a certain volume, a static grammar becomes absolutely necessary to provide an accessible documentation for anyone who has not participated in developing the program. Moreover, such a grammar is essential in a modular strategy, when a team of several people working with relative autonomy write the modules of a dynamic grammar; such being the case in the industrial development of a linguistic model.