A CONTEXT-FREE GRAMMAR OF FRENCH*

Morris Salkoff
Université de Paris 7, L.A.D.L.
2 Place Jussieu, Paris 5, FRANCE

Summary. I present here a method that allows one to construct a CF grammar of a natural language that correctly accounts for verbal selection rules. This goes contrary to the prevailing opinion, following Chomsky[5], that such a construction is impossible. My method consists essentially in separating the semantic function of a selection rule, namely, the exclusion of certain noun sub-classes, from the syntactic relation between the elements (verb, subject, object) linked by this relation of selection. When the verb and object (subject) are separated by intervening levels of complement construction, the selection can still be satisfied by a double classification of verbs: according to the kind of subject they take, and also according to the type of verb that can follow them (in the complement construction). Conjunctions and sentences with *respectively* can also be treated within the framework of the CF approximation proposed here.

§0. Introduction

It is now quite generally supposed that a natural language cannot be adequately described by a CF grammar. This opinion was first advanced by Chomsky[5] who discussed this problem from the point of view of phrase structure grammars. He presents there a fragment of a CF phrase structure grammar in terms of noun phrases NP, verb phrases, VP, etc., which are familiar from immediate constituent analysis. These rules cannot treat verbal selection rules properly; Chomsky[4] (ch. 8) had already tried himself to correct this defect within the framework of a CF phrase structure grammar, but the difficulties he encountered seem to have persuaded him that only a transformational grammar could handle such a problem.

Harman[13] proposed another solution to the problem of treating verbal selection rules in a CF grammar; he added a set of subscripts to the CF rules used in Chomsky[5], which were chosen so that only those subjects and objects which satisfied the selection rules could appear with a given type of verb. Chomsky[8] showed that this method would not suffice if the sentences subscripted as Harman had suggested were themselves embedded in complement constructions. Thus, where Harman's system will not generate such aberrant sentences as **Bill elapsed*, it will not be able to exclude the generation of such a sequence when it is embedded in a complement construction, as in **John persuaded Bill to elapse*.

Further arguments for the inadequacy of a CF grammar were adduced from the fact that sentences containing *respectively* cannot be assigned an appropriate structure in the framework of a CF grammar. This was noted by Chomsky[5] (§4.2) in his discussion of the algebraic language $\underline{w}\ \underline{w}$; the relation between this language and sentences containing *respectively* was discussed by Bar-Hillel & Shamir[1], and then taken up again by Chomsky[6] together with examples taken from the comparative construction in English. Later, Postal[22] exhibited a construction in Mohawk which is similar to the one with *respectively*, and like the latter, is recursively extendable to sentences of theoretically unbounded length.

As a result of these considerations, Chomsky[8] concluded that a coherent description of recursively embedded sentences or of verbal selection rules could not be obtained in a natural way by any CF grammar, and that consequently no CF grammar could adequately describe a natural language. However, it turns out that this question is not so easily disposed of as it would appear, and recent work by Joshi & Levy[18] shows that a CS grammar containing rich context-dependent rules can be used to analyze trees that describe a CF language. They did this by an extension of a theorem of Peters & Ritchie[21], who showed that CS rules of a certain type can be used not to generate sentences, i.e., not to characterize them, but only to verify their well-formedness, by applying the context-dependent parts of these rules as constraints on the set of trees that schematize these sentences. In this case, the language described by these trees is a CF language.

Joshi & Levy generalized the kinds of CS rules that can be used for this result and defined CS rules that can describe conditions on the context whose action is close to that of certain transformations. These rules are expressed as Boolean combinations of predicates that describe the left and/or right context of a node, or the upper and/or lower contexts (the nodes above and below a given node). Roughly speaking, a tree is said to be analyzable with respect to a grammar containing such rules if one of the rules is satisfied at each node of the tree. In that case, the language which consists of the terminal strings of all the trees analyzed by the grammar is a CF language, even though the rules take the context into account§. Hence these terminal strings can be described by

§Note that the formalism used by Joshi & Levy for displaying conditions on trees is close to the notation used for rewrite rules, and can lead to some confusion. It need only be remembered that these context-dependent rules are not used to generate structures.

some CF language.

Now the string grammar proposed by Harris[15] and which analyzes English (Sager[23]) and French (Salkoff[24,25]) can be shown to be of just the form described by Joshi & Levy. It contains CS rules of the type described by them, and is used to analyze a tree, rather than to generate it. It would thus appear that English or French can be described by some CF language, **although the** string grammar gives no clear clue as to what its form would be. I shall show here that such a CF grammar can be written for French, and that it can treat, in a linguistically appropriate fashion, the problem of the expression of verbal selection rules in nested complement constructions. I have chosen French because systematic data giving a wide coverage of the French lexicon are available (Gross[12], Boons et al.[3]); however, the very nature of this construction makes quite plausible its extension to other natural languages. Only the method used will be outlined in this brief article, and an example of its application to embedded complement constructions; for more details, consult Salkoff[25] (chap. 3).

I now construct CF rules that correctly describe sentences in which related pairs like verb-subject, verb-object, etc., that are linked by a relation of selection, may be separated by constructions of unbounded length. Each such CF rule is the expansion of a sentence schema S. The verbal selection rules are accounted for in this method by separating the semantic function of a selection rule, namely, the exclusion of certain noun sub-classes, from the syntactic relation between the pairs carrying this function (generally, a verb and a noun phrase). Each selection rule is decomposed into two independent parts: one part is the choice of a noun not classified in certain noun sub-classes, in such a way as to express the semantics of that selection rule; the second part is the use of the noun phrase containing this $\underline{N}$ for the subject or object of a given verb in $\underline{a}$ rule schema, which amounts to satisfying the complete verbal selection rule.

Conjunctional sequences, including sentences containing *respectivement* can be handled by this method, but not within the strict mathematical framework of a CF language. The resulting CF grammar of French can be compared with a transformational grammar, and it is seen that the two are more similar than has been thought.

§1. The base rules

In order to set forth the selection rules as clearly as possible, I shall begin by using in the rules developing S, noun phrases bearing three subscripts, i.e., complex symbols:

(1) $NP_{x,y,z}$; where $\underline{x}$ is a function $\underline{F}$: subject $\underline{s}$, object $\underline{o}$, or indirect object $\underline{io}$; $\underline{y}$ is the morphology $\underline{M}$: singular, plural,...; and $\underline{z}$ is a semantic sub-class S; these sets have no elements in common.

With this notation, typical rules for S will have the following form:

(2) a $S \rightarrow NP_{s,y,z}\ t\ V_i\ NP_{o,y,z}$

b $S \rightarrow NP_{s,y,z}\ t\ V_j\ P\ NP_{io,y,z}$

c $S \rightarrow NP_{s,y,z}\ t\ V_k\ NP_{o,y,z}\ P\ NP_{io,y,z}$, etc.

The verb is subscripted according to the complements it takes.

In this notation, the CF rules no longer constitute a strict constituent grammar of the type discussed by Chomsky[5,8]. My notation brings out the grammatical relations between the elements of the sentence schemata, which is not possible in a direct way in a phrase structure grammar.§ The complex symbols are useful in order to explain clearly the process of sentence embedding; they will be eliminated in a second step and replaced by the noun phrases without subscripts used in the verbal selection rules.

Main rule schema.

For clarity, I shall use only the subscript $\underline{F}$ ($\underline{s}$, $\underline{o}$, or $\underline{io}$) in the rules for S. Only an abbreviated list of these rules can be given here; for a complete list, cf. Salkoff[25]. A first subgroup of rules contains non-sentential objects:

(3) $S \rightarrow NP_s\ t\ V_1$ (*Max dort*)

$S \rightarrow NP_s\ t\ V_2\ NP_o$ (*Max signe le traité*)

$S \rightarrow NP_s\ t\ V_3\ P_i\ NP_{io}$ (*Paul dépend de Max*)

$S \rightarrow NP_s\ t\ V_4\ NP_o\ P_i\ NP_{io}$ (*Paul base sa théorie sur ces faits*); etc.

There are about ten such rules in French. A second group of rules contain a sentential complement clause:

$S \rightarrow NP_s\ t\ V_{20}$ *que* S (*Max sait que Paul a fait cela*);

$\rightarrow NP_s\ t\ V_{22}\ NP_o$ *que* S (*J'informe Max que Paul est venu*)

A third group of rules yield embedded sentences. One example will be treated here, as it occurs in independent sentences and in relative clauses, to illustrate the method.

(4) a $S \rightarrow NP_s\ t\ V_{30}\ S^1$ (*Max convainc ...*

$S^1 \rightarrow {}_oNP_s$ *de* V_1 *... Paul de dormir*)

$\rightarrow {}_oNP_s$ *de* $V_2\ NP_o$ *..Paul d'oter cela*)

The new notation ${}_oNP_s$ denotes a noun phrase having a double function $\underline{F}$: it must be an accept-

---

§To do so, one has, for example, to reinterpret the tree structure of the sentence (cf. Chomsky[7]).

able object of the verb V_{30} which precedes§, and also an acceptable subject of the main verb of S^1. The sentence schema for S^1 is a sentence deformation (in Harris'[17] terminology); there are about ten such deformations in French. Another one is the following:

(5) $S \rightarrow NP_s\ t\ V_{32}\ S^3$ (*Max apprend ...*

$S^3 \rightarrow à\ {}_{io}NP_s\ à\ V_1$ ... *à Paul à dormir*)

... etc.

Each such schema S^i contains as many rules as S itself.

With the schemata S^i, I can account for the recursive embedding of sentences, like *Luc convainc Paul d'apprendre à Max à dire aux élèves que...*; other schemata are needed to account for sentence embedding in relative clauses:

(6)a $NP^{r1} \rightarrow {}_{o}NP_s$ *que* $NP_s\ t\ V_{30}\ S^1_{pro}$ (*l'homme que Max convainc...*

b $S^1_{pro} \rightarrow {}_{o}(\emptyset)_s$ *de* V_1 .. *de dormir*)

$\rightarrow {}_{o}(\emptyset)_s$ *de* $V_2\ NP_o$ .. *d'oter cela*)

. . . etc.

Here, the symbol ${}_{o}(\emptyset)_s$ is a dummy element standing for the noun phrase, carrying the same subscripts, at the head of NP^{r1}. It is marked by the same selectional features as ${}_{o}NP_s$ and will be used to 'transmit' this selection through embedded sentences. Such dummy elements come close to certain pronouns found in relative clauses without antecedent, like *ce* in: *J'ai acheté ce que Max a sculpté.*

A second type of relative clause is this:

(7) $NP^{r2} \rightarrow NP_o$ *que* S_{pro} (*le livre que...*

$S_{pro} \rightarrow NP_s\ t\ V_2\ (\emptyset)_o$ ... *Paul lit*)

$\rightarrow NP_s\ t\ V_4\ (\emptyset)_o\ P\ NP_{io}$.*Paul fait de ces articles*)

. . .

$\rightarrow NP_s\ t\ V_{30}\ S^{1.1}_{pro}$ .. *Max convainc...*

$S^{1.1}_{pro} \rightarrow {}_{o}NP_s$ *de* $V_2\ (\emptyset)_o$ .. *Paul de lire*)

$\rightarrow {}_{o}NP_s$ *de* $V_4\ (\emptyset)_o\ P\ NP_{io}$

..*Paul de faire de ces articles*)

. . .

$\rightarrow NP_s\ t\ V_{32}\ S^{3.1}_{pro}$ .. *Max apprend...*

$S^{3.1}_{pro} \rightarrow à\ {}_{io}NP_s\ à\ V_2\ (\emptyset)_o$ .. *à Paul à lire*)

. . .

. . . etc.

With these rules, it is possible to describe recursively embedded sentences inside relative clauses, although the complex symbols give us no indication yet as to how the verbal selection rules are to be satisfied.

§2. Selection Rules

According to the kind of noun allowed as subject, or as direct or indirect object, a verb is said to select for that sub-class. The majority of the selection rules thus concern the following three rules for S:

(8)a $S \rightarrow NP_s\ t\ V_2\ NP_o$ (*Luc porte un chapeau*)

b $S \rightarrow NP_s\ t\ V_3\ P_i\ NP_{io}$ (*Max dépend de Luc*)

c $S \rightarrow NP_s\ t\ V_4\ NP_o\ P_i\ NP_{io}$ (*Max attribue la médaille à Luc*)

The selection rules vary with the preposition P_i for verbs V_3 and V_4. In the sentence analyzer based on the string grammar, these selection rules are contained in a system of contextual rules attached to each lexical entry for a verb that can appear in (8). Experience shows that five noun sub-classes are needed for such a system of selection rules: N_t, 'time'; N_s, sentential; N_h, 'human'; N_c, concrete; and N_{nom}, nominalizations. These sub-classes are used in the verbal entries to indicate the unacceptable contexts for a verb classified in V_2, V_3, or V_4. The analyzer then uses these contextual rules to disallow an unacceptable decomposition in a sentence analysis.

These contextual rules can be replaced by CF rules in the following way. Let

(9) $N \setminus \{N_i + N_j + \ldots\} = N^\smallsetminus$

denote any noun except one belonging to subclass N_i, or to N_j, etc.; the bar \ means 'minus'. If $N^\smallsetminus$ is substituted for the noun N in any NP, and carried over into every rule developing NP, the terminal rule for the noun in NP will be

(10) $N^\smallsetminus \rightarrow N_d$, where N_d is a lexical entry.

Each choice for N_d is compared with the list of sub-classes $N_i, N_j, \ldots$, attached to $N^\smallsetminus$. If N_d belongs to any of these sub-classes, it is discarded; if N_d doesn't belong to these sub-classes the conditions expressed in (9) are satisfied. Now, if the selection rule of a given verb is that sub-classes $N_i, N_j, \ldots$, are unacceptable as subject (object), then the noun phrase containing N_d satisfies that selection rule, and will be the only noun phrase permitted in that syntactic position.

I now define noun phrases GN containing all the combinations of excluded noun classes from the five named above (there are 31 such GN):

(11)a $GN \rightarrow N$, if no sub-classes are excluded;

b $GN_1 \rightarrow N\setminus\{N_t\}$; $GN_2 \rightarrow N\setminus\{N_s\}$; ... ;

$GN_5 \rightarrow N\setminus\{N_{nom}\}$;

c $GN_{1,2} \rightarrow N\setminus\{N_t + N_s\}$; $GN_{1,3} \rightarrow N\setminus\{N_t + N_h\}$

$\ldots GN_{1,5} \rightarrow N\setminus\{N_t + N_{nom}\}$;

§Sentences like *Max empêche que la table ne tombe → Max empêche la table de tomber*, in which the raised object (*table*) does not have to be compatible with the verb *empêcher*, are accounted for by different rules.

$GN_{2,3} \to N\backslash\{N_s + N_h\}; \ldots$

$\vdots$

Inserting the noun phrases $GN_{i,j}$ into (8), and replacing the subscripts i and j by the single subscript j, I obtain the following rule schemata:

$$(12)\quad S \to (GN_j)_s\ t\ V_{j,j'}\ (GN_{j'})_o\ ; \quad 1\le j,j'\le 31$$

$$S \to (GN_j)_s\ t\ V_{j,j''}\ P_i\ (GN_{j''})_{io} \quad 1\le j,j''\le 31$$

$$S \to (GN_j)_s\ t\ V_{j,m}\ (GN_{j'})_o\ P_i\ (GN_{j''})_{io} \quad 1\le m\le (j'\times j''\times k)$$

The subscripts are not independent; in general, a verb accepts a certain $GN_{j'}$ ($GN_{j''}$) only for certain values of GN_j. This is captured in the double verb classification: $V_{j,j'}(V_{j,j''})$ is that verb sub-class which requires GN_j for subject, when the direct (indirect) object is $GN_{j'}$ ($GN_{j''}$). Lexicographical work shows that there are about 40 different prepositions appearing in the objects P N and N P N. Since the double verb classification must be carried out for each value of P_i, this amounts to a triple classification of verbs.

§3. Elimination of the complex symbols

The schema (12) generates only acceptable sentences; each verb in the lexicon is classified according to which of the sub-classes defined by (12) it belongs to; hence no verb will ever appear in a schema of type (12) unless it is acceptable there. Then, since the process defined by (10) is such that only acceptable nouns can be chosen for the noun phrases GN (= N`) in these schemata, each schema must in fact give rise to an acceptable sentence.

The situation is quite different, however, for the rules containing $_oNP_s$, NP_o or NP_{io}. These cannot be developed as written, for two reasons: (1) only noun phrases of the type N` are available, so that verbal selection rules can be satisfied; (2) the syntactic functions expressed by the subscripts on these noun phrases can be obtained only by a sub-classification of the verbs appearing with them. Thus, in order for $_oNP_s$ (in 4) to be an acceptable object of the verb V_a that precedes, and also an acceptable subject for the verb V_b of the embedded sentence containing it, the verb V_b must be sub-classified according to type of subject, and V_a has to be sub-classified according to the type of V_b that may follow.

An even more complex classification is needed to handle relative clauses like (7), which begin with NP_o. This noun phrase must be an acceptable object for the last verb, say V_c, in the S_{pro} which follows; however, S_{pro} can contain an unbounded number of embedded verbs before V_c appears. Hence, V_c is not known at the moment when the lexical entry is chosen for the N` which represents NP_o. The problem, then, is to 'transmit' the selectional characteristics of the noun in NP_o to the rule that will later develop V_c, by using the embedded verbs as carriers for the selectional information. This transmission of selectional information necessitates a sub-classification both of embedded verbs and of the schemata of the type S_{pro}.

§31 Elimination of $_oNP_s$

I subdivide the sets S^1, S^2, S^3, ... (cf.4 and 5) into S^1_i subsets, where i runs through the 31 possible values of the subject N` (which replaces $_oNP_S$). These subsets then constitute a classification of the schemata S^1, .., according to the type of subject that is acceptable for the verb of the schema:

$$(13)\quad \begin{array}{ll|ll} S^1_1 & \to N`_1\ de\ V_1 & S^1_2 & \to N`_2\ de\ V_1 \\ & \to N`_1\ de\ V_2\ NP_o & & \to N`_2\ de\ V_2\ NP_o \\ & \ldots & & \ldots\text{etc.} \\ & \to N`_1\ de\ V_{30}\ S^1_i & \vdots & \\ & \to N`_1\ de\ V_{31}\ S^2_j & S^1_{31} & \to N`_{31}\ de\ V_1 \\ & \ldots & & \ldots \end{array}$$

I subdivide the sets S^1_{pro} (6b), S^2,..., in the same way:

$$(14)\quad S^1_{pro,1} \to (\emptyset)_1\ de\ V_1, \text{ where } \emptyset_1 \text{ is a dummy carrying the selectional features of } N`_1;$$

$$\to (\emptyset)_1\ de\ V_2\ NP_o$$

$$\ldots$$

$$\to (\emptyset)_1\ de\ V_{30}\ S^1_{pro,j}$$

$$\ldots \text{ etc.}$$

This new way of ordering the rules is the basis for the sub-classification of verbs V_{30}, which take the object S^1. A verb V_{30} accepts only the sub-sets S^1_i whose subject $N`_i$ is an acceptable object for that verb. This is a selection rule between verbs: the verb V_{30} selects an object having a verb of a certain type†.

The generation of recursively embedded sentences which satisfy verbal selection rules is now obtained as follows. First, let us choose a rule developing the matrix sentence, for example

$$(15)\quad N`_j\ t\ V_{30}\ S^1_i$$

Now the verbs in the sub-class V_{30} have been subclassified in the lexicon according to the type of acceptable subject, $N`_j$, and also according to the type of acceptable complement $S^1_i$. By choosing in (15) a verb in the sub-class $(N`_j, S^1_i)$, I obtain an acceptable sentence.

†The selection between verbs mentioned here has already been suggested by Z. **Harris**[16] in the framework of a system of sentence generation based on the concept of the verb as an operator acting on its arguments (approximately, its subject and object). Selection between verbs was also used by M. Gross[10] in order to account for constructions like *Je cours manger un gâteau*, ?? *Je cours détester Max*; here, the first verb (of movement) selects for the type of verb that can follow it.

Next, S^1_i is developed, using the schema (13), by one of two types of rules:

(16) a $S^1_i \rightarrow N^{`}_i$ *de* $V_{31}\ S^2_j$; b $S^1_i \rightarrow N^{`}_i$ *de* $V_2\ NP_o$

If rule a is chosen, another sentence is embedded, and a verb V_{31} in the sub-class $(N^{`}_i, S^2_j)$ is chosen from the lexicon. But if rule b is chosen sentence embedding terminates with that rule.

The same method can be used for generating acceptable relative clauses NP^{r1} (in 6). As an example, I rewrite one of the NP^{r1} in terms of the noun phrases $N^{`}$:

(17) a $NP^{r1} \rightarrow N^{`}_i$ *que* $N^{`}_j\ t\ V_{30}\ S^1_{pro,i}$

b $S^1_{pro,i} \rightarrow (\emptyset)_i$ *de* V_1

By choosing a verb V_{30} in the subclass $(N^{`}_j, S^1_{p,i})$ i.e., one taking $N^{`}_j$ as its subject and as second verb (in S^1_{pro}) one whose subject is $N^{`}_i$, I guarantee that the $N^{`}_i$ in a is both an acceptable object of V_{30} and an acceptable subject of the verb in S^1_{pro}.

§32 Elimination of NP_o

The development sketched in §31 will not do for relative clauses like NP^{r2} (in 7), which have the form NP_o *que* S_{pro}. This can be schematized roughly as NP_o *que*. . V_i . . V_j .. V_c , where V_i, V_j, .. are embedded verbs of the type V_{30}, V_{31}, ..., and V_c is the last verb of S_{pro}, the one for which NP_o must be an acceptable object.

In order to transmit the selectional characteristics of NP_o to the rule that develops V_c, and this within the framework of a CF grammar, I can proceed as follows. I subscript S_{pro} in (7) by k, which is also the subscript on the noun phrase $N^{`}_k$ that replaces NP_o (just as S^1, S^2,..., were subscripted for the type of subject); then the schema $S^{1.1}_{pro}$ for embedded sentences will have two subscripts: one for k, and a second one for the type of subject the verb takes. This yields the following kind of development:

(18) $NP^{r2} \rightarrow N^{`}_k$ *que* $S_{pro,k}$

$S_{pro,k} \rightarrow N^{`}_i\ t\ V_2\ (\emptyset)_k$

$\rightarrow N^{`}_i\ t\ V_4\ (\emptyset)_k\ P_n\ NP_{io}$

. . .

$$
\begin{array}{l}
31 \\ \text{sche-} \\ \text{mata}
\end{array}
\left\{
\begin{array}{l|l}
\rightarrow N^{`}_i\ t\ V_{30}\ S^{1.1,k}_{pro,1} & \rightarrow N^{`}_i\ t\ V_{31}\ S^{2.1,k}_{pro,1} \\
\rightarrow N^{`}_i\ t\ V_{30}\ S^{1.1,k}_{pro,2} & \rightarrow N^{`}_i\ t\ V_{31}\ S^{2.1,k}_{pro,2} \\
\quad . . . & \quad . . . \\
\rightarrow N^{`}_i\ t\ V_{30}\ S^{1.1,k}_{pro,31} & \rightarrow N^{`}_i\ t\ V_{31}\ S^{2.1,k}_{pro,31}
\end{array}
\right.
$$

$S^{1.1,k}_{pro,1} \rightarrow N^{`}_1$ *de* $V_2\ (\emptyset)_k$

$\rightarrow N^{`}_1$ *de* $V_4\ (\emptyset)_k\ P_n\ NP_{io}$

. . .

$\rightarrow N^{`}_1$ *de* $V_{30}\ S^{1.1,k}_{pro,j}$

$\rightarrow N^{`}_1$ *de* $V_{31}\ S^{2.1,k}_{pro,j}$

. . .

(31 times, as above, once each for the subjects $N^{`}_2, \ldots, N^{`}_{31}$)

The typical rule for NP^{r2} is the following:

(19) $NP^{r2} \rightarrow N^{`}_k$ *que* $N^{`}_i\ t\ V_{30}\ S^{1.1,k}_{pro,j}$

Once more, acceptability is guaranteed by choosing a verb V_{30} in the sub-class $(N^{`}_i, S^1_j)$. Next, the symbol $S^{1.1,k}_{pro,j}$, representing a possibly embedded sentence, can be developed by the rules:

(20) a $S^{1.1,k}_{pro,j} \rightarrow N^{`}_j$ *de* $V_{32}\ S^{3.1,k}_{pro,j'}$,

b $S^{1.1,k}_{pro,j} \rightarrow N^{`}_j$ *de* $V_2\ (\emptyset)_k$

If rule b is chosen, sentence embedding terminates; then, choosing a V_2 in the sub-class taking an object of type $N^{`}_k$ (as indicated by $\emptyset_k$) guarantees that $N^{`}_k$ in (18) is an acceptable object for that V_2. If rule a is chosen, sentence embedding continues; a verb V_{32} is chosen, in the sub-class $(N^{`}_j, S^3_{j'})$, until a rule of type b is chosen.

The reader will notice two features of this method of using the selection rules to generate relative clauses.

(1) The subdivision of S^1 into a set of S^1_i rule schemata does not increase the number of rules in S. The same number of rules would be obtained by inserting the noun phrases $N^{`}_k$ into S (or S^1), and this must be done in any case in order to express the verbal selection rules (in whatever fashion). In the decompositions of S^1,..., used above, the point was only to present the original schemata so as to make the subject or object of the verb in the schema stand out, for further reference.

(2) The two kinds of selection made explicit in these schemata, the one between verbs, and the other (better known) between verb and object (or subject), appear only once in the grammar. Both types of selection are used in each step of sentence embedding, but in no case does this entail rewriting the two kinds of selection in the grammar each time a deeper level of embedding is attained.

§4. Conjunction; *respectively*

It has been shown by Chomsky[5] that conjunctions can be described in a CF grammar only by using an infinite number of rules, represented by rule schemata; if one restricts oneself to strict CF grammar, one introduces an excessive structuring of the conjoined forms. An approximate solution can nevertheless be given to this problem, in the framework of a finite CF grammar, in the following way. I construct a sequence of conjoined noun phrases:

(21) a $GN^1 \rightarrow N^{`}$ ; b $GN^2 \rightarrow N^{`}$ *et* $N^{`}$;

c $GN^3 \rightarrow N^{`}$ *et* $N^{`}$ *et* $N^{`}$;

d $GN^i \rightarrow N^{`}$ *et* $N^{`}$ ... *et* $N^{`}$ (i times)

Denoting by G^i_{cf} the CF grammar containing the rules GN^i, GN^{i-1}, ..., GN^1, I can set up the series of grammars G^1_{cf}, G^2_{cf}, ..., G^i_{cf}, each representing a better approximation to the infinite grammar G^{∞}_{cf}, which contains a noun phrase of unbounded length.

For any practical purpose, such as generation (or analysis) of sentences, it is clear that one of the G^i_{cf} will be large enough to yield the desired precision. However, another approximation is available which is less costly, from the viewpoint of the number of rules required, and which yields the same result for G^i_{cf}. This is the rule schema proposed by Chomsky & Schützenberger[9] for handling conjunction in a CF grammar. For the case of noun phrase conjunction, this schema is as follows:

(22) a $GN \rightarrow N`$; b $GN \rightarrow N`$ $(et\ N`)^*$

The star indicates that the group $(et\ N`)$ can be iterated as many times as is necessary. This schema is therefore an abbreviation for an infinite number of rules.

With such a rule schema in it, my grammar is no longer strictly CF; however, it is clearly faithful to the spirit of the approximation for G^i_{cf} outlined above, since the language described by my grammar is the same as that reached asymptotically by the series of grammars G^1_{cf}, G^2_{cf},..., G^i_{cf} obtained with (21). The rule schema (22) can be compared to an algorithm for generating any one of the grammars G^i_{cf} by choosing the number of iterations.

There exists a set of structures in natural language which cannot be described by the methods developed until now, namely those containing either *respectivement*, or the distributives *qui* or *selon que*:

(23) a *Les rats des groupes A et B réussissent et échouent dans les labyrinthes L_a et L_b, respectivement.*
b *Les reporteurs ont parlé qui aux ministres, qui aux délégués, qui aux députés.*
c *Selon que tu es pauvre, bourgeois ou aristocrate, tu seras ouvrier, commerçant ou patron.*

Although these strings cannot be generated by a CF grammar§ , a procedure is nevertheless available for including this type of sentence in the CF approximation under discussion here.

§The applicability of this argument to the linguistic case is not quite as simple as this brief formulation of the argument might lead one to suppose, in the way it is generally used in discussing sentences with *respectivement*. It is only the language containing just the sentences (23), and only those, that cannot be generated by a CF grammar. However, in order for this conclusion to apply to the generation of the entire French language by a CF grammar, it must be shown that there exists no sublanguage of French containing these sentences in *respectivement* as a subset that can be generated by a CF grammar. Cf. Gross[11] (§8.1) for this argument.

I add Kleene rules to the grammar, and a condition on these rules, as follows:

(24) a $N_s\ (et\ N_s)^*\ V\ N_o\ (et\ N_o)^*$
b $N_s\ (et\ N_s)^*\ V\ (et\ V)^*\ N_o\ (et\ N_o)^*$

These rules contain all common conjunctions of subject, verbs and direct object. Moreover, they cover the sequences of classes observed in sentences containing *respectivement*. They don't have the structure one would like to associate with such sentences. In order to describe the *respectivement* sentences, I add the following condition to the starred parentheses: the number of iterations of each occurrence of the star is the same; and a structure, or rule of interpretation, is imposed on the starred groups, as follows:

(25) $N_s\ (et\ N_s)^*\quad V\ (et\ V)^*\quad N_o\ (et\ N_o)^*$

This grouping pairs the N_s and the N_o that are to be associated with each other via *respectivement*; (25) is equivalent to:

(26) $N^1_s\ et\ N^2_s\ ..\ V^1\ et\ V^2\ ..\ N^1_o\ et\ N^2_o$

Thus, I am interpreting (25) as a sentence conjunction: $N^1_s\ V^1\ N^1_o\ et\ N^2_s\ V^2\ N^2_o\ et$ ..., as required by the adverb *respectivement*.

§5. Conclusions

The methods I have sketched here can certainly be applied to other natural languages and will account in a natural way for the general phenomena of verbal selection rules in embedded sentences. One may wonder why this work has not been carried out before.

Historically, attacks against the adequacy of CF grammar for describing natural language arose at a moment when it was necessary to explore the nature of the transformational grammar just proposed. This new style of grammar seemed so much better adapted than CF phrase structure grammars to explaining sentence relations that any more effort towards developing a detailed CF grammar seemed fruitless. To discourage such efforts, Chomsky[5] (chap. 5) declared that "any grammar that can be constructed in terms of this theory [CF phrase structure grammar] will be extremely complex, *ad hoc* and 'unrevealing'". These remarks were reaffirmed (Chomsky[8]) and bolstered by an argumentation based on the inherent inadequacy of CF grammar for describing verbal selection rules.

A second criticism arose from the analysis of constructions, like *respectivement*, whose description could not be obtained within the strict framework of a CF grammar. We have seen above that such a statement is at best unclear. It may be correct that a mathematically rigorous description of this construction is not possible in a strict CF grammar; even so, we are under no obligation to transfer this observation bodily to the domain of linguistics. The type of description that I elaborated above, in which a rule of interpretation is added to a rule generating the form of sentences containing *respectivement*, is now used in recent work in generative semantics.

Moreover, it can be seen that the CF grammar presented here is but a short step removed from a transformational grammar. In all transformational theories, a transformation includes (among other things) a relation between sentences. Most authors also include operations that deform one sentence into another, or which modify an abstract structure so as to derive sentences from it. The CF grammar I have proposed contains the information that establishes relationships between sentences, but it does not contain the operations or the metalinguistic assertions that make the transformation explicit. By a small extension of the CF framework I can also obtain the equivalent of a transformation, as follows. As an example, I consider the passive transformation.

The passive transformation consists in matching an active phrase with its passive counterpart. The statement of the transformation can stop there, as does Harris'[14], or one can add the specification of the computer operations needed to create the active and passive trees, as in generative grammar. In the CF grammar presented here, I have two independent rules, one for the active form, and another for the passive of the first:

(27) a $S_{act} \rightarrow NP_s\ t\ V_2\ NP_o$

b $S_{pas} \rightarrow NP_o\ t\ \textit{être}\ V_2\acute{e}\ (\textit{par}\ NP_s)$

Each of these rules has an independent set of selection rules that are expressed in the choice of the $N^{`}$ for the NP. Adding these selection rules, (27) becomes:

(28) a $S_{act} \rightarrow NP_s\ t\ V_2\ NP_o\ ;\ NP_s \rightarrow N^{`}_i;\ NP_o \rightarrow N^{`}_j$

b $S_{pas} \rightarrow NP_o\ t\ \textit{être}\ V_2\acute{e}\ (\textit{par}\ NP_s)\ ;$
$NP_o \rightarrow N^{`}_j\ ;\ NP_s \rightarrow N^{`}_i$

This is of course a wasteful repetition of identical selection rules. It was just to avoid this kind of useless duplication that justified the introduction of transformations. Suppose now that I factorize the selection rules from a set of forms that constitute an equivalence class, for example, from the 'active' and the 'passive' forms; I place a separator ρ between the forms of the equivalence class:

(29) $S \rightarrow NP_s\ t\ V_2\ NP_o/\rho/\ NP_o\ t\ \textit{être}\ V_2\acute{e}\ (\textit{par}\ NP_s)/\rho/\ \textit{Il}\ t\ \textit{être}\ V_2\acute{e}\ NP_o\ (\textit{par}\ NP_s);$
$NP_s \rightarrow N^{`}_i\ ;\ NP_o \rightarrow N^{`}_j$

In this formulation, the selection rules are no longer duplicated; moreover, we can interpret the separator ρ between the members of the equivalence class as indicating a relation between the sentence schemas so separated. The factorization of the selection rules, together with the introduction of the separator can be read as the definition of a transformational rule between the sentence schemata.

Of course, rule (29) is no longer CF, but it represents a rather natural extension of the CF framework which makes the latter much more similar to a transformational grammar than one might have thought possible up til now. However, the reader will note that the concept of a transformation is indispensable as a tool for the construction of this CF grammar, and then for its extension towards a transformational grammar by means of the factorization of selection rules. Furthermore, this CF grammar does not generate the sentences of the language 'weakly', in the meaning given this word by Chomsky; in fact, it provides them an 'adequate' grammatical structure as well as a linguistically justifiable relationship to other sentences of the language.

Finally, let us note that although the entire set of rules of the CF grammar proposed here is large (of the order of 10^9 rules), it is nonetheless finite. Furthermore, the size of the grammar is of no theoretical consequence, since it could be stored quite handily, not in some static memory (e.g., a pile of discs), but in a dynamic form (that is, in the form of schemata) where each rule is generated at the moment when the program of syntactic analysis (or generation) requires it. In this way, the set of rules would be reduced to a series of sub-programs that can generate either one rule, or a sub-set of rules, or all the rules. During analysis or generation, a call for rules would activate their *synthesis* by the appropriate sub-program.

Such a program of analysis by synthesis reduces the number of rules to a smaller number of sub-programs, but a string grammar reduces them still more, down to a set of about 150 strings (the rewrite rules) together with about 200 restrictions (the CS portions attached to the CF rules).

The size of the CF grammar required to describe selection rules adequately also explains why all attempts at automatic syntactic analysis by means of strictly CF grammars undertaken until now have failed. The authors of these CF grammars limited their effort to including some rudimentary linguistic facts; the average size of this sort of CF grammar was of the order of several thousand rules (cf. Kuno[19,20]). Under these conditions, there was no question of providing only linguistically acceptable analyses. However, in the last few years, other CS variants of a CF grammar have been proposed, and partly worked out. In particular, the augmented transition network grammar of Bobrow & Fraser[2], especially in the form given it by Woods[26], has predicates associated with the transitions, predicates that are so many context-sensitive tests. This kind of grammar is then quite similar to string grammar, i.e., to a CF grammar together with CS conditions on the rules. Unfortunately, none of the grammars based on the ideas of Bobrow and Woods has been worked out in sufficient detail to make a linguistic comparison with string grammar possible.

BIBLIOGRAPHY

1. Bar-Hillel & Shamir,E., 1960. Finite-State Languages, in *Language & Information*, New York, Addison-Wesley, (1964).

2. Bobrow, D. & Fraser, B., 1969. An augmented state transition network analysis procedure, *Proc. of the International Joint Conference on Artificial Intelligence*.

3. Boons, J.-P., Guillet, A. & Leclere, C., 1976. *Classes de constructions transitives*, Rapport de Recherche N° 6, L.A.D.L., Univ. de Paris 7, Place Jussieu, Paris

4. Chomsky, N., 1955. *The logical structure of linguistic theory*, New York, Plenum (1975).

5. —— 1957. *Syntactic Structures*, The Hague Mouton

6. —— 1963. Formal properties of grammars, in *Handbook of Mathematical Psychology*, Vol. 2, New York, John Wiley

7. —— 1965. *Aspects of the theory of syntax*, Boston, MIT Press

8. —— 1966. Topics in the theory of generative grammar, in *Current Trends in Linguistics*, Vol. 3, The Hague, Mouton

9. Chomsky, N. & Schützenberger, M., 1963. The algebraic theory of context-free languages, in *Computer Programming and Formal Systems*, Amsterdam, North-Holland

10. Gross, M., 1968. *Grammaire transformationnelle du français: le verbe*, Paris, Larousse

11. —— 1972. *Mathematical Models in Linguistics*, New Jersey, Prentice-Hall

12. —— 1975. *Méthodes en Syntaxe*, Paris, Hermann

13. Harman, G., 1963. Generative grammars without transformation rules: a defense of phrase structure, *Language*, Vol. 39, N° 4.

14. Harris, Z. 1952. Discourse analysis, *Language*, Vol. 28, N° 1

15. —— 1962. *String analysis of sentence structure*, The Hague, Mouton

16. —— 1964. The elementary transformations, in Harris, 1970, *Papers in structural and transformational linguistics*, Dordrecht, Reidel

17. —— 1968. *Mathematical structures of language*, New York, John Wiley

18. Joshi & Levy, 1977. Constraints on structural descriptions: local transformations, *SIAM J. of Computing*, Vol. 6, N° 2

19. Kuno, S., 1963. The multiple-path syntactic analyzer for English, Report N° NSF-9, Computation laboratory, Harvard, Boston.

20. —— 1965. The predictive analyzer and a path elimination technique, *Comm. of the Assn. for Comp. Mach.*, Vol 8, p. 453

21. Peters, S. & Ritchie, R., 1969. Context-Sensitive immediate constituent analysis, *Proc. of the ACM Symposium on 'Theory of Computing'*, New York, ACM

22. Postal, P., 1964. Limitations of phrase structure grammars, in *The structure of language*, ed. by Fodor & Katz, New Jersey, Prentice-Hall.

23. Sager, N., 1973. The string parser for scientific literature, in *Natural Language Processing*, ed. by R. Rustin, New York, Algorithmics Press.

24. Salkoff, M., 1973. *Une grammaire en chaîne du français*, Paris, Dunod

25. —— 1979. *Analyse syntaxique du français: grammaire en chaîne*, Amsterdam, J. Benjamins

26. Woods, W., 1970. Transition network grammars for natural language analysis, *Comm. of the Assn. for Comp. Mach.*, Vol. 13, p. 591

*I should like to thank M. Gross for many helpful comments, and myself for an excellent typing job.