

A PROGRESS REPORT ON  
THE USE OF SLANT GRAMMAR CALCULUS FOR  
AUTOMATIC ANALYSIS

Ferenc Kiefer

The KVAL-approach to syntactic analysis has been based on categorial grammar as propounded by Bar-Hillel and Lambek. It was clear at the very outset that the original form of categorial grammar is not suitable for the purpose of automatized syntactic analysis in practice. The first attempts at developing a more adequate notation as well as at the closer examination of various properties of categorial grammar go back to 1965 and 1966 /KVAL PM 237, 248, 298, 302/. In order to distinguish our approach from original categorial grammar we have termed it slant grammar calculus.

Our practical concern has been to write a grammar for Swedish nominal phrases which would enable us to automatize some linguistic aspects of documentation research. Such a grammar has been compiled by Bengt Svensson. The rules utilize lexical and morphological information. In other words, no strict line is drawn between morphological and syntactic rules. A substantial part of this preliminary grammar has already been checked generatively by means of a special string processing

algorithm /STRIP/.

As known one of the most tedious problems in connection with categorial grammar is the great number of categories assignable to most of the units in the string to be analyzed. This is a par excellence practical problem but it does not lack theoretical implications. The most interesting of these can be roughly phrased as follows: If syntax is to be kept as simple as possible, then the lexicon will get complicated. If we aim at the optimal simplification of the lexicon, then the burden of our grammar will be taken over by syntax. (Recall that in categorial grammar we have only two rules for the bidirectional case. Therefore, all possible syntactic functions for each lexical entry must be listed in the lexicon.) Now we may ask questions about the optimal distribution of tasks of a grammar between syntax and lexicon in order to achieve an overall simplification in grammatical description. (Notice that the same question can be asked with respect to syntax and semantics, or still better, with respect to syntax, semantics and the lexicon.) This problem has, of course, not yet been solved and consequently, one is forced to work in general with several alternatives.

On the other hand, one can neglect this theoretical issue and concentrate on strategies that would lead to an essential reduction of the possible ambiguities for a given string. Some proposals to this effect can be found in KVAL PM 327 and 373. One of these strategies follows the usual path: the point is being made

that the establishment of the correct syntactic structure /or structures/ should be carried out in several /at least two/ steps.

Thus, instead of having rules of the form

$$\begin{array}{l} a/b \quad b \rightarrow a \\ /1/ \\ a \quad a \backslash b \rightarrow b \end{array}$$

we write rules like

$$/2/ \quad x \circ y \rightarrow z$$

which may mean either

$$z/x \quad x \rightarrow z$$

or

$$x \quad x \backslash z \rightarrow z$$

In /2/ we neglect the dominance relation /see below/ holding between the two constituents and also their linear order. For the latter point notice that /2/ corresponds to two phrase structure-type rules:

$$\begin{array}{l} z \rightarrow a + b \\ /3/ \\ z \rightarrow b + a \end{array}$$

where + stands for concatenation.

As to the dominance relation it stands to reason to interpret  $z/x$  in a string  $z/x \quad x$  as being the head or governor of the construction. Similarly, for a string  $x \quad x \backslash z$  we would say that  $x$  is governed by  $x \backslash z$ . In view of the way the cancellation rules of the slant grammar calculus are built up we may say that the

"more complex" or "longer" category is the governor of a given expression. This information is clearly lost in /2/.

However, as specified in Interim Report No 11, we are not forced to make this interpretation. The slant calculus as such lends itself to phrase structure grammar too, context-free or context-sensitive.

Instead of /1/ our grammar would now contain a set of statements of the form /2/:

$$\begin{array}{l} a_1 \circ b_1 = c_1 \\ /4/ \quad a_2 \circ b_2 = c_2 \\ \quad \dots \\ \quad a_n \circ b_n = c_n \end{array}$$

Each statement in /4/ corresponds to a syntagm type in the given grammar. Of course, it is not necessary to have binary rules only. One might also have

$$a_1 \circ a_2 \circ \dots \circ a_m = c_1$$

etc

Now a string of categories in terms of our slant grammar calculus can cancel to a "simpler" category if, and only if, one of the rule schemata in /4/ holds. But a set of such schemata will yield many different categorial grammars, varying as to the interpretation of dependency.

With the help of /4/ one can determine a sort of "deep" structure for a given sequence of morphemes.

After having determined this deep structure we can proceed by finding out more about the actual structure.

Another approach consists of combining the aforementioned multi-level method with probabilistic considerations. For details see KVAL Interim Report No 12.

Since we have interpreted categories in terms of dependencies it is readily seen that slant grammar calculus can be considered as a dependency-type grammar. The dependency statement for the rules /1/ can be rendered as /5/:

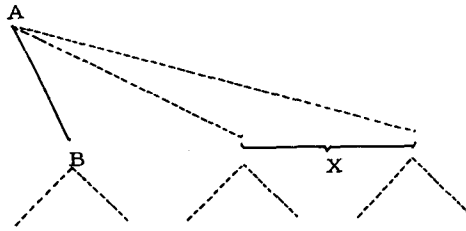
$$\begin{aligned} a/b & ( * [b] ), \\ a \backslash b & ( [a] * ) \end{aligned}$$

We fully subscribe to the view that the determination of the head or governor of a construction is an important and indispensable task. /See, also, Jane Robinson/ Therefore, we think that dependency grammar in whatever notational conventions it is expressed, is superior to phrase structure grammar. Jane Robinson has argued that many transformations need a reference to the head of construction which in usual phrase structure grammar can only be provided by some ad hoc device. All considerations with respect to dependency grammar hold with equal force with respect to slant grammar calculus. The latter can be used as a base for transformations in the same way as dependency grammar can. Following Robinson we may make use of the following notational conventions that differentiate dependency structures from phrase structures: we may use asterisks to mark governing occurrences, parentheses

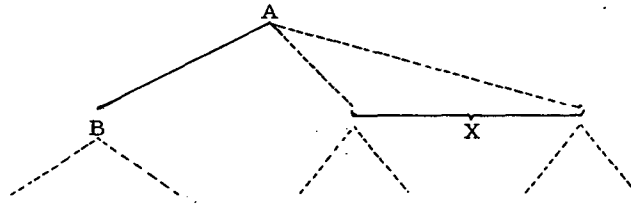
to mark boundaries and a special pair of symbols /labels/, denoting a variable depth of nesting. For example,

$$(A^* B X) \Rightarrow \begin{matrix} 2 & 1 & 3 \\ 1 & 2 & 3 \end{matrix}$$

In other words, the transformation rule applies to a family of trees with the structure



and as a result we get a family with the structure



/Robinson, op.cit. pp. 26-27/.

It is now of little importance what the symbols A, B and X denote, i.e. whether they denote categories characteristic of dependency grammar or of slant grammar calculus. We can thus conceive of a grammar that has as its base /"categorical component"/ a dependency-type grammar which is context-free and a transformational component that operates on structures generated by the categorical

component. In this case we can expect from slant grammar calculus a notational advantage at best.

In fact, this seems to be the case. Thus, in many cases one can take advantage of the resemblance between categories of the slant grammar calculus and ordinary fractions. Under certain conditions we can determine the type of syntagm by assigning to each symbol in the sequence of categories a prime number and then carry out the cancellation in an arithmetical sense. Then, sequences of category symbols in the slant grammar calculus can easily be handled without any reference to their meanings. Several proposals have been worked out that take advantage of exactly this trait /KVAL PM 367/. One definement is proposed in Interim Report 5, where relatively prime 2 x 2 matrixes with integer elements are assigned to the atoms of any categorial grammar, so that a string is grammatical if, and only if, the product of the assigned matrixes is equal to some "unit".

On the other hand, we may ask to what extent transformational rules are really necessary for our purpose. Could they not all be replaced by context-sensitive rules? This problem has been examined in some detail in Interim Report 11. Another kind of categorial symbols, with selectors as well as numerators and denominators and the cancellation rule /5/ were proposed in order to cover the context-sensitive case instead of /1/:

$$\begin{array}{l}
 /5/ \quad x // y \quad y \rightarrow x \quad y \\
 \quad \quad y \quad y || x \rightarrow y \quad x
 \end{array}$$

and in "mixed" cases:

/6/            u z z | u || x / y // v v y → u x v

Some formal questions concerning the various grammars in the framework of slant grammar calculus have been tackled in

H. Karlgren: Multi-index Syntactic Calculus.

So far it is not clear whether rules of type /5/ or /6/ will solve all our problems. But it is in this direction that we want to work next.



### References

1. Kiefer, Ferenc: Bestimmung der syntaktischen Konnexität von Morphemesequenzen, KVAL PM 237 /1965/
2. Kiefer, Ferenc: Ein Algorithmus für Konnexitätsbestimmung, KVAL PM 248 /1965/
3. Kiefer, Ferenc: Lexical Redundancy Rules in Categorical Grammar, KVAL PM 302 /1966/
4. Kiefer, Ferenc: The Question of Ambiguity in Categorical Grammar, KVAL PM 298 /1966/
5. Brodda, Benny: Om sekvenser av matriselement, KVAL Interim Report No 5 /1967/
6. Kiefer, Ferenc: The Possibility and/or Necessity of CS-rules in Categorical Grammar, KVAL Interim Report No 6 /1968/
7. Karlgren, Hans: Slant Grammar Calculus, KVAL PM 367 /1967/
8. Karlgren, Hans: Unique Labelling, KVAL PM 373 /1968/
9. Karlgren, Hans: Categorical Grammar Analysis of Context-Sensitive Languages, KVAL Interim Report No 11 /1968/
10. Karlgren, Hans: Lösande av kategorialuttryck, KVAL Interim Report No 12 /1968/
11. Karlgren, Hans: Multi-index Syntactic Calculus, to appear in Computational Linguistics, Vol VIII
12. Robinson, J. Jane: Dependency Structures and Transformational Rules, IBM, Thomas J. Watson Research Center, Scientific Report No 3, Yorktown Heights /1968/