# IMPLEMENTING THE GENERALIZED WORD ORDER GRAMMARS OF CHOMSKY AND DIDERICHSEN

by Bengt Sigurd, Dept of Linguistics, Lund University, SWEDEN
Helgonabacken 12, S-223 62 Lund, e- mail: linglund@gemini.ldc.lu.se

Many of the insights of Transformational Grammar (TG) concern the movability of constituents. But in recent versions (Government & Binding, GB; cf. Chomsky, 1982, Sells,1985) the sentence representations (trees) include both the site of the moved constituent and the site from where it has been moved; the original site of the moved constituent is marked as a trace (t) or empty (e, []). In the sentence schema (Field or Position Grammar) developed by the Danish linguist Paul Diderichsen (1946), there are also positions both for the new and the old site of moved constituents. Thus Diderichsen observes that an adverb could introduce or be the fundament of a sentence, in which case the subject np "remains" in its "normal" position after the finite verb (Swedish example: *Idag kom pojken*; literally: Today came the boy). If the subject np introduces the sentence (*Pojken kom idag*) its "original" place after the finite verb must be empty (For comparisons between Transformational Grammar and Diderichsen's grammar, cf. Teleman, 1972, Platzack,1986).

Underlying both Chomskyan GB grammar and Diderichsen's Field Grammar is a grammatical system which consists of a general word or constituent order schema supplemented with co-occurrence restrictions. This type of system may be called Generalized Word Order Grammar (GWOG), and this paper deals with ways of implementing such a system on the computer using Definite Clause Grammar (DCG;Clocksin & Mellish, 1981), a formalism available in most Prolog versions.

Definite Clause Grammar is a convenient rewriting system with an arrow (-->) familiar to generative linguists. It allows one to state the maximum sequence of constituents (the order schema) to the right of the arrow. A setup of constraining conditions can then be used to prohibit overgeneration. Such restrictions are stated within curly brackets in the Definite Clause Grammar formalism. Constraining conditions may require that certain slots be filled or empty, that a certain variable have a certain value, that certain constituents cannot occur at the same time (co-occurrence restrictions), etc.

In addition one may have further conditions which state that a certain constituent is to have a certain functional role, e.g. be the subject or the object of the sentence. Such conditions may be called functional role conditions (f-conditions) as they build a functional structure (f-representation). This structure may be built in a certain slot (as an additional argument) to the left of the arrow. Further conditions may concern the topic (focus), mode, clause type, lacking constituent, etc. of the sentence, and this information may also be gathered as arguments in slots to the left of the arrow.

The system to be presented in this paper also incorporates many of the ideas of Referent Grammar (RG; Sigurd, 1987), a functional generalized phrase structure grammar used in the automatic translation project Swetra (Sigurd & Gawronska-Werngren, 1988). I hereby acknowledge the help of Mats Eeg-Olofsson,

Barbara Gawronska-Werngren and Per Warter in the Swetra group at Lund.

## The generalized word order schemas of Chomsky and Diderichsen

As can be seen from articles and text-books (e.g. Sells,1985), a typical Chomskyan Government & Binding representation is a high binary hierarchical tree with complementizer phrases (C-phrases) on top of I(nfl)- and V-phrases. A tree for the Swedish sentence: "Vem slog pojken?" (Literally: Whom hit the boy?) given here in a parenthesis notation might look as follows, assuming "pojken" ("the boy") to be the subject:

```
CP(XP(vem:i),
    C'(C(slog:j),
        IP(NP(pojken:k),
            I'(I,VP(NP(e:k),
                V'(V(e:j),
                    NP(e:i)))))))
```

"Whom hit the boy"

This simplified representation means that the object "vem" is found in a front slot called "XP", the finite verb is found in the slot called "C(omplement)" and the subject "pojken" is found in the "specifier" slot under IP. The "spec" under "VP" is empty and so are the verb slot under V' and the NP slot under V'.

The transformational (process) description would say that "vem" ("whom") has been moved from its final position leaving a trace indexed with the same number (e:i) for reference. Similarly the transformational description would say that the finite verb "slog" and "pojken" have left coindexed traces (e:j, e:k) behind. The Swedish sentence: "Vem slog pojken" is ambiguous and could also be interpreted as "Who hit the boy". In that case the question pronoun "vem" (now equivalent to English "who") should be coindexed with a trace in the position where "pojken" was found in the first case and "pojken" should be found in the "object position" under V'.

Diderichsen uses a simpler model - he did his work long before Chomsky when formal grammar was not as highly developed. He would have stated the facts in the following way:

| **Fund** | **v** | **s** | **a** | **V** | **S** | **A** |
|---|---|---|---|---|---|---|
| Vem | slog | pojken | - | - | - | - |
| Vem | slog | - | - | - | pojken | - |

For the first interpretation of the sentence the "object slot" S(ubstantive=nominal) is empty; for the second interpretation the subject slot s(ubstantive) is empty - besides the empty slots for sentence adverbs (a), non-finite verbs (V) and other adverbs (A) also marked by the minus sign (-). Diderichsen calls the first three slots "the nexus field" and the last three "the content field" (indholdsfeltet). This division suits sentences containing an auxiliary with infinitives or participles, but for other sentences the division between a nexus field and a content field is unfortunate. The objects (in S) get separated from the finite verb (v) in simple transitive sentences. In the model to be presented below infinitives and participles are treated as subordinate (minor) clauses with their own objects and adverbs.

## GWOG rules - a simple illustration

The following (simplified) Prolog (Definite Clause Grammar) rules illustrate how examples like those mentioned in the introduction can be handled by Generalized Word Order Grammar rules.

```
sent(M,T,
s(subj(Subj),pred(Pred),advl(Advl)))-->
 fund(Fund),vi(V),np(Np2),adv(Adv2),
    {Fund=np(_),Np2=[],Subj=Fund,
       Pred=V,Advl=Adv2,T=Fund,M=d;
    /*Subj+Verb+Adverb:Pojken kom idag*/
       (Fund=adv(_),Np2\=[],Subj=Np2,
       Pred=V,Advl=Fund,T=Fund,M=d}.
    /*Adverb+Verb+Subj:Idag kom pojken*/
```

This basic rule is a rewriting rule. It states that we get the information in the argument slots after "sent" if we find the (phrase or word) categories to the right of the arrow in the order they are given. Further phrase and word (lexical) rules defining an adverb (adv), an np, and an intransitive verb (vi), e.g. as described in Sigurd(1987) are needed. The lexical rules needed in order to generate our examples can have the following simplified form: np(np(pojken)) --> [pojken]. np([]) --> []. vi(kom) --> [kom]. adv(adv(idag)) --> [idag]. adv([]) --> []. The categories np and adv may be empty ([]). The verb is obligatory. Diderichsen's "fundament" ("fund") is an initial position unspecified as a syntactic category. Both an np and an adverb may occur as fundament in our simple example, so the following two fundament rules are therefore needed:

fund(F) --> np(F). /* an np is fundament */
fund(F)--> adv(F). /* an adv is fundament */

As can be seen, the schema would be overgenerating if no co-occurrence restrictions were introduced. Such restrictions or conditions are written within curly brackets ({}) in Definite Clause Grammar, and they state which conditions are to hold on the variables specified. (Variables begin with capital letters in Prolog.)

Two alternatives are shown with examples. The first alternative occurs if the fundament is an np: np(_,[Fund],[]). In that case no second np (Np2) can be found after the intransitive finite verb. (This is our way of stating that an np has been fronted). In addition to the co-occurrence restrictions, the sample rules illustrate how information about functional roles and topic is stated. In the first case the fundament (Fund) is assigned the functional role of subject. The value of the fundament is also assigned to the Topic variable (T).

In the second alternative, given after semicolon (;), an adverb is the fundament: adv(_,[Fund],[]). Then there must be an Np2 (Np2 cannot be empty: Np2\= []). In that case the subject is assigned the value (Np2) and the adverb (Fund) is the topic of the sentence. The value of the adverb (Fund) is also assigned to the adverbial (Advl) of the functional representation. In both cases the Pred is assigned the value (V) of the verb, and in both cases the mode of the sentence is declarative, why M(ode) is set at d(eclarative). The two examples would both receive the following functional representation:

s(subj(pojken),pred(kom),advl(idag)).

This functional representation agrees with the standard format of Referent Grammar used in machine translation. The order in an RG functional representation is fixed: subject, predicate, dative obj, direct object, sentence adverbials, other adverbials.

As can be seen there are slots for Mode, Topic and Functional representation with "sent". The output of the parsing of a sentence

is information about mode, topic and the functional representation. In more advanced and extensive rules, information about clause type and defectiveness (in order to handle the percolation of missing constituents) is also gathered in additional slots with "sent".

## A generalized word order schema for Swedish

Generalizing from the word and constituent orders found in Swedish one may suggest the following basic rule for main clauses:

```
sent(M,Cl_type,Defect,T,F_repr) -->
fund(Fund),    idag      []      pojken
v(V),          kom       gav     lovade
sadv(Sadv2),   inte
np(Np2),       pojken    pojken  flickan
sadv(Sadv3),             inte
np(Np3),                 flickan
prediv(Prediv),
np(Np4),                 hunden
sunt(Sunt),                      att gå
adv(Adv2).               idag
```

The Swedish examples to the right show how slots may be filled differently: "Idag kom inte pojken" (Literally: Today came not the boy), "Gav pojken inte flickan hunden idag?" Literally: Gave the boy not the girl the dog today?),"Pojken lovade flickan att gå" (Literally: The boy promised the girl to go). "Sunt" is the category containing subordinate clauses and minor (infinitive or participial) clauses.

Compared to Diderichsen's model there is a longer sequence of categories, and non-finite verbs are treated as subordinate clauses. Chomsky and his followers try to define functional roles configurationally, but our approach is rather a formalization of Diderichsen's verbal descriptions. The functional representation is built as a list in the more advanced versions, but we will not go into such technical details here.

The following are further illustrations of the conditions needed:

```
{Fund=[],vtt(V),Np2\=[],Np3\=[],Np4\=[],
 subj(Np2),dobj(Np3),obj=Np4,M=q;
 /* gav pojken flickan hunden? */
Fund=np(_),vd(V),Np2\=[],Np3=[],Np4=[],
Sunt\=[],subj(Fund),dobj(Np2),obj(Sunt),
M=d}. /* pojken lovade flickan att gå */
```

The first condition states that if there is nothing (Fund=[]) before a doubly transitive finite verb (vtt), the mode must be "q(uestion)" and the noun phrases are assigned the roles: subject, dative object (dobj) and direct object (obj) in that order. This covers our example "Gav pojken (inte) flickan hunden idag?" (Literally: Gave the boy (not) the girl the dog today?"). The second alternative (after ;) shows the case of "verba dicendi" (vd) as in "Pojken lovade flickan att gå" (Literally: The boy promised the girl to go). In that case the first noun phrase after the finite verb (Np2) is taken as a dative object and the infinitive clause represented by "Sunt" as the direct object.

## Discussion and conclusion

It is clear that there is a trade-off between the extension (generality) of the order schema and the co-occurrence restrictions. A very general schema requires many constraining restrictions, several simpler schemas require fewer restrictions, but the overall system grows bigger. Chomsky and his followers seem to prefer to use one schema to cover all types of clauses in order to catch as many generalizations as possible. The node name

"comp(lementizer)" clearly stems from subordinate clauses, but it has been generalized to all sentences in GB. Diderichsen used one general schema for all types of main sentences, but a separate schema for subordinate clauses. For a general discussion of the potential of positional systems in syntax, morphology and phonology see Brodda & Karlgren, 1964.

Some of our restrictions and constraints on the value of certain variables and co-occurrence of constituents, etc. can be related to the constraining principles and filters used in GB.

Swedish subordinate clauses differ from main clauses by having the sentence adverbs before the finite verb, and generally subordinate clauses are characterized by initial complementizers, such as subjunctions, infinitive markers or relative pronouns. In the current implementation subordinate clauses are treated by separate rules. In Swedish, almost all information about clause type, topic, and mode is to be found in the positions before the finite verb.

It is clear that the GWOG model suits the Nordic and Germanic languages well with their finite verb second and fairly fixed word order, but not languages with fairly free word order (e.g Slavic languages) where the schema must allow for almost any combination of the words.

The program illustrated works nicely for analysis, but when used for synthesis (generation) further conditions are needed and the components have to be rearranged somewhat. The program may be considered as an alternative to Pereira's Extraposition grammar (1981).

**References**


B. Brodda & H. Karlgren, 1964. Relative positions of elements in linguistic strings. SMIL 3, 49-101

N. Chomsky, 1982. Some concepts and consequences of the theory of government and binding. Cambridge, Mass: MIT Press

W.Clocksin & C. Mellish, 1981. Programming in Prolog. Berlin: Springer

P. Diderichsen, 1946 (3:rd ed). Nudansk grammatik. København:Gyldendal

F. Pereira, 1981. Extraposition grammar. American Journal of Computational Linguistics 7,4, October-December 1981

Chr.Platzack. 1986. Diderichsens positionsschema och generativ transformationsgrammatik. In: Heltoft & Andersen (eds). Saetningsskemaet og dets stilling - 50 år efter. Nydanske studier 16-17, Roskilde: Akademisk forlag

P. Sells, 1985. Lectures on contemporary syntactic theories. Stanford: CSLI

B. Sigurd, 1987. Referent Grammar. A generalized phrase structure grammar with built-in referents. Studia Linguistica 41:2,115-135

B. Sigurd & B. Gawronska-Werngren, 1988. The potential of SWETRA - a multilanguage MT-system. Computers andTranslation 3, 238- 250

U.Teleman, 1972. Om Paul Diderichsens syntaktiska modell. In: Tre uppsatser om grammatik. Lund: Studentlitteratur