*Carl Bache**

Presentation of a pedagogical sentence analysis system

1. Introduction

The sentence analysis system that I want to present in this paper[1] is the result of work that began back in the early days of the English Department at Odense University, Denmark, in the mid-seventies, when the late Professor Hans Hartvigson decided to fight a noble battle against the widespread ignorance of basic grammatical facts among new generations of young and hopeful English students – students who claimed never to have heard of the pluperfect, or the subjunctive, students who did not know the subtle difference between 'predicate' and 'predicative', and who did not cherish the vital distinction between a noun and a subject, between form and function. This dismal state of affairs has changed, mostly thanks to Professor Hartvigson. He was the prime mover in setting up a new introductory course in grammar, and in preparing suitable teaching materials (cf. Hartvigson et al. 1977). With Leif Kvistgaard Jakobsen, John Dienhart, Fritz Larsen and others, he started the work on sentence analysis that eventually led to the publication in 1991 of *An Introduction to English Sentence Analysis* by Mike Davenport, John Dienhart, Fritz Larsen and myself - an exercise book which has now appeared in a second, revised (1993) edition. An interactive program for computer aided instruction in this approach to sentence analysis was devised as early as 1983 (cf. Davenport et al. 1984). While the system may naturally be challenged theoretically, it

1 I would like to thank Niels Davidsen-Nielsen, John Dienhart, Peter Harder, Leo Hoye, Fritz Larsen, Christian Heyde Petersen and Marianne Stølen for useful comments on earlier presentations of the system and the concept of 'stacking'.

* *Carl Bache*
Institute of Language and Communication
Odense University
Campusvej 55
DK-5230 Odense M

works very well pedagogically: our students actually learn to analyse sentences as a prerequisite to reading and understanding standard university grammars, and some of them even think it is good fun!

In the following I shall offer a presentation of a slightly modified version of our sentence analysis system, the version that Professor Niels Davidsen-Nielsen and I are using in a new grammar project funded by the Danish Research Council (cf. Bache & Davidsen-Nielsen 1995). I would like also to review some of the characteristic features of our system in a broader syntactic context, especially because it differs from current syntatic theory and practice in a number of important ways. Finally I shall introduce the concept of 'stacking' which I find very useful in the analysis of some challenging types of examples which the system cannot handle unless modified in some way.

2. An elementary sentence analysis system.

Let me begin by offering an introduction to our approach to sentence analysis. We draw a basic distinction applying to all levels of analysis between the *form* and the *function* of constituents. We have at our disposal a reasonably small number of both form terms and function terms for the description of each and every constituent. The Appendix provides an overview of all the terms used in our system. Let me briefly draw attention to some of the more important ones. We recognize five basic sentence functions:

[1]	S	=	subject	*Jack left*
	P	=	predicator	*Jack left*
	O	=	object	*Jack left his wife*
	A	=	adverbial	*Jack left her last year*
	C	=	complement	*Jack was pretty foolish*

We operate with four main types of form:

[2]	w	=	single word	*Jack*, *left*
	g	=	group of words	*his wife*, *last year*, *pretty foolish*
	cu	=	compound unit	*The old man and his sister left*
	cl	=	clause	*The old man and his sister left*

Single words are further specified according to word class as follows:

[3]	n	=	noun	*car, letter, party, idea*
	v	=	verb	*write, be, receive, hear*

adj	=	adjective	*long, old, afraid, big*
adv	=	adverb	*slowly, gently, again*
pro	=	pronoun	*he, she, who, any, this*
prep	=	preposition	*by, at, to, from, in*
conj	=	conjunction	*that, because, if*
art	=	article	*the, a(n)*

In addition there are interjections (intj), numerals (num) and the infinitive marker (infm). Notice the use of capital letters for functions and lower-case letters for forms.

The two types of relation 'subordination' and 'coordination' are specified in this way:

[4] Subordination:

SUB	=	subordinator	*(He said <u>that</u> he liked her)*

[5] Coordination:

CO	=	coordinator	*(The old man <u>and</u> his sister)*
CJT	=	conjoint	*(<u>The old man</u> and <u>his sister</u>)*

To show the illocutionary value of an utterance we replace the top label, 'Sent' (which represents the sentence as a whole), with the following labels for utterance functions:

[6] Main utterance functions:

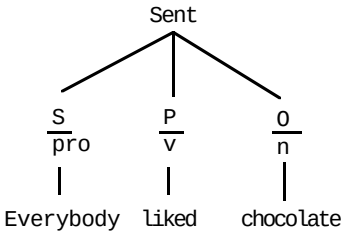
STA	=	statement	*John left*
QUE	=	question	*Did John leave?*
DIR	=	directive	*Get out!*
EXC	=	exclamation	*How wonderful you are!*

For our analysis of a constituent we select both a function term and a form term. We employ two different techniques in our structural representations: *linear analyses* and *tree diagrams*, exemplified in [7] and [8]. The two techniques are purely notational variants, i.e. different ways of showing the same structure. In a linear analysis (which is convenient for simple or partial analyses in run-on texts), we use square brackets to indicate the beginning and the end of constituents. The function term and the form term are separated by a colon, with the function term to the left of the colon and the form term to the right. In [7] there is a linear analysis of a simple sentence:

[7] Linear analysis of *Everybody liked chocolate*:
S:pro$_{[everybody]}$ P:v$_{[liked]}$ O:n$_{[chocolate]}$

The tree diagram, which is a conventional form of plotting syntactic relationships within a sentence, offers a nice overview of an analysis, as in [8]. The lines slanting downwards from the uppermost constituent indicate a 'consist-of' relationship. Instead of the colon convention, we use the 'function-over-form' convention. Thus the tree diagram in [8] should be read in this way: we have a sentence which consists of a subject realized by the pronoun *Everybody*, a predicator realized by the verb *liked* and an object realized by the noun *chocolate*.

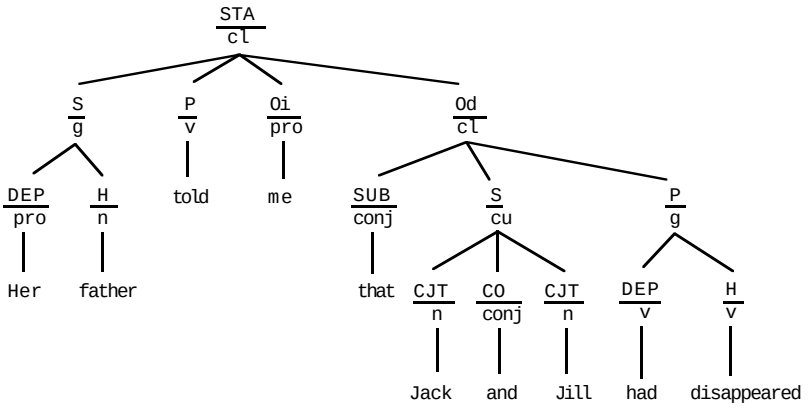
[8]



The interesting thing about form and function is that there is no one-to-one relationship between them: given the function of a constituent you cannot predict the form, and vice versa. With few exceptions, any function can be realized by any form type, and any form type can assume any function.

Let us look at another, more complex example:

[9]



This sentence is a statement realized by a clause (STA:cl). The clause consists of a subject, a predicator, an indirect object and a direct object. All main types of form are present in [9]: for example, the predicator and the indirect object are realized by single words, the subject of the main clause and the predicator of the object clause are realized by groups, the direct object is realized by a clause, and the subject of the direct object clause is realized by a compound unit. Each of the sentence functions in [9] could have been realized by other form types. For example, the single pronoun *I* might have been subject instead of the noun group *Her father*. The verb group *had told* could have replaced the single verb *told.* The indirect object pronoun *me* could have been replaced by, say, a compound unit *Sally and Bob*. And instead of having a direct object clause, we might have had just a noun group *the truth*. What all this goes to show is that there are in principle few restrictions on what combinations of form and function are possible.

Similarly, virtually any illocutionary value can be realized by any form type (cf. Bache et al. 1993:205ff, Bache & Davidsen-Nielsen 1995:103ff):

[10]	STA:cl	*James left Brisbane yesterday.*
	STA:g	*In London.*
	STA:cu	*John and Sarah.*
	STA:intj	*Yes.*
[11]	QUE:cl	*Will you join me tonight?*
	QUE:g	*From whom?*
	QUE:cu	*When and where?*
	QUE:adv	*Why?*
[12]	DIR:cl	*Listen to me.*
	DIR:g	*After him!*
	DIR:cu	*Smile and be happy.*
	DIR:adv	*Down!*
[13]	EXC:cl	*What a fool he is!*
	EXC:g	*Good Lord!*
	EXC:cu	*Damn and blast!*
	EXC:intj	*Wow!*

When an utterance function is realized by a clause, it is possible to subclassify the clause *formwise* as declarative (decl), interrogative

(inter), imperative (imp) or exclamatory (excl). Though these clausal form types relate in an obvious way to the illocutionary functions (as we see in [10] to [13] above), there is no strict one-to-one correspondence. For example, the declarative *James left Brisbane yesterday* will be interpreted as having the illocutionary value QUE if pronounced with rising intonation.

In short, our system is a fairly simple but flexible one which will cope nicely with a good many ordinary utterances and written sentences.

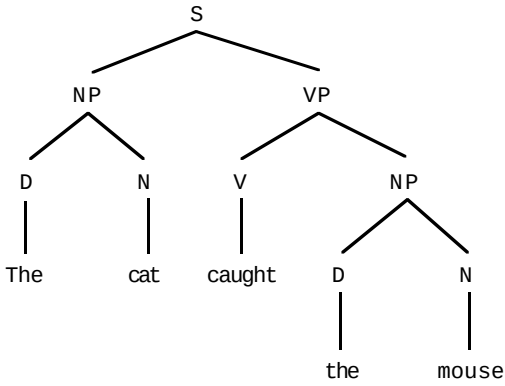
3. Special features

While our system is pedagogically attractive, it differs considerably from mainstream syntax in a number of important ways:

A) No transformations. Note first of all that we do not operate with transformations or conversions of structures of any kind. Our analysis yields a relatively pure form of surface structure. When I say 'relatively pure' it is because we occasionally allow 'missing forms' in examples like: *The agent (that) we met last night / I parked my car behind John's (car) / Jack became president and Ann (became) vice-president / She told us (that) she wanted to come*, cf. Bache et al. 1993:85ff, Bache & Davidsen-Nielsen 1995:81ff).

B) No predicate. Note next that we do not operate with a predicate. Traditionally, the predicate is a super-function, or metafunction, comprising the predicator plus all subsequent functions in the clause. For example, in *Everybody liked chocolate* (cf. [8] above), the predicate consists of the predicator verb *liked* and the direct object noun *chocolate.* In *Her father told me that Jack and Jill had disappeared* (cf. [9] above), the predicate contains the predicator verb *told*, the indirect object pronoun *me* and the direct object clause *that Jack and Jill had disappeared*. In traditional syntax, the primary syntactic division of a sentence is into subject and predicate. This division is simply taken for granted by most modern syntacticians. For example, in early transformational syntax, a sentence like *The cat caught the mouse* would be analysed as in [14] (see e.g. Jacobs & Rosenbaum 1968):

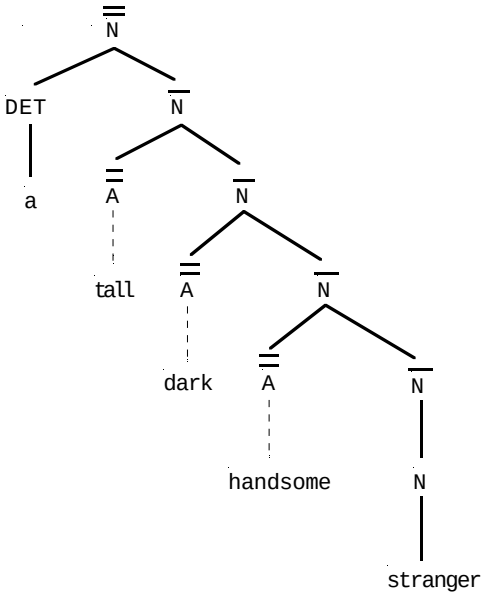
[14]



In this analysis the predicate is marked as a VP which contains the verb *caught* and the noun phrase *the mouse*. In our sentence analysis we do not normally divide the sentence into subject and predicate but simply skip the predicate level.

C) Multibranching. The third characteristic of our system that I would like to mention is that it allows complex constituents to divide into more that just two daughter constituents. In our view, constituents are potentially *multibranching.* The binary principle, to which many syntacticians are attracted, is of course nice and simple, and it is very much in fashion these days in computer and digital technology. However, we believe language to be potentially more complex. The unbearable lightness of the binary principle is clear if you consider e.g. Radford's X-bar analysis of the noun group *a tall dark handsome stranger* in [15] (cf. Radford 1980:104):

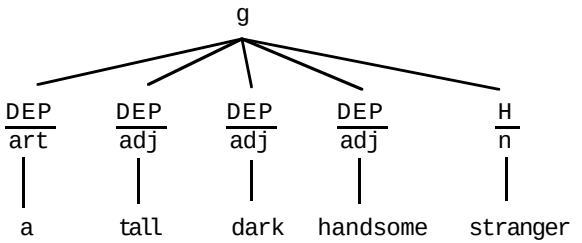
[15]



(X-bar analysis provides a notation system for marking different levels of group-like structures within groups: double-bars indicate a superordinate level, single-bars intermediate levels, and 'no bars' the terminal lexical level.)

In our system a possible analysis of this noun group is offered in [16]:

[16]



In this particular case I think it is clear that our analysis is preferable to Radford's: the three adjectives seem to be paratactically related, i.e. to be at the same level in relation to *stranger.* Note that we do not change the meaning of the noun group if we separate the adjectives with commas and/or *and*:

[17] She was visited by a tall dark handsome stranger
= She was visited by a tall, dark and handsome stranger

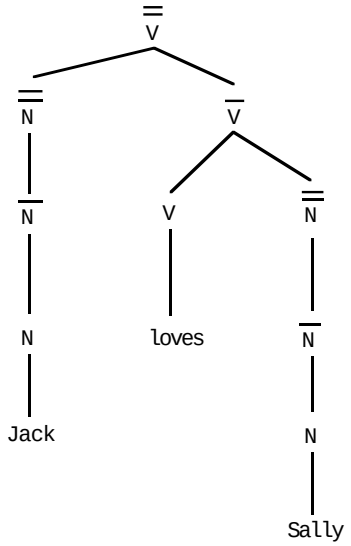
In other noun groups, we have hypotaxis, as in the following example:

[18] The envious Republican senators complained
≠ The envious, Republican senators complained

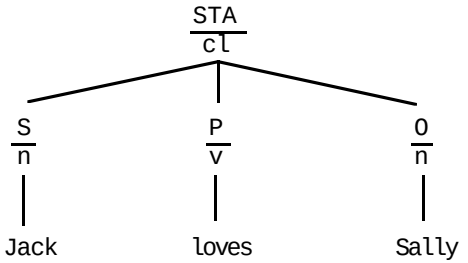
Without the commas, *envious* either offers a description of the Republican senators present in the context or it describes a subclass of the Republican senators: some but not all the Republican senators are envious. In other words, *envious* either describes or subcategorizes *Republican senators*. By contrast, when *envious* and *Republican* are separated by a comma, the two adjectives relate individually to the head noun *senators.* The implication here is rather that the envious senators present in the context happen to be also Republicans. If we want to be able to reflect this subtle difference between hypotaxis and parataxis, one way of doing it is to have both multibranching and binary branching as options. Note that our recognition of multibranching is not simply a question of 'maximal bracketing' in *notational* opposition to 'minimal bracketing' (cf. Halliday 1994:20ff), though this opposition is a valid consideration. We claim that certain structures cannot be represented appropriately by means of binary branching.

D) No non-branching constituents. In our system, unlike most other systems, there are no non-branching constituents, except at the terminal word level, i.e. the level at which lexical insertion takes place. It is interesting to compare a typical X-bar analysis of e.g. *Jack loves Sally* in [19] with our analysis of the same sentence in [20]:

[19]



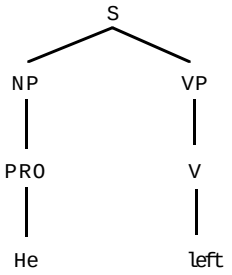
[20]

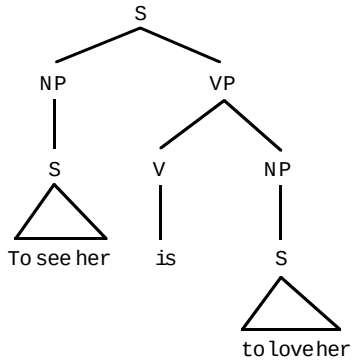


Instead of saying that we do not operate with non-branching nodes we can say that we automatically prune non-branching nodes. Pedagogically, it is a relief not having to explain why *Jack* and *Sally* are constituents at three different levels, as in the X-bar analysis in [19] (N, N single bar and N double bar). In our model, we say things more directly: *Jack* is a noun with subject function and *Sally* is a noun with object function.

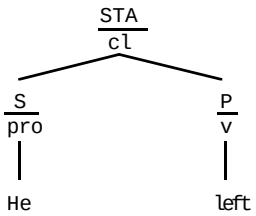
E) What is a group? The next feature that I would like to mention here concerns the identity and nature of groups. As will be clear from the analyses offered so far, we take a rather unusual stand on this issue. We say that a group is a group. A group is not a word or a clause. What is more, we say that the head of the group determines its subclassification: if the head of a group is a noun, the group is a noun group; if the head of a group is a verb, the group is a verb group, if the head of a group is an adjective, the group is an adjective group, and so forth. This is all very unorthodox. In classical generative transformational grammar, to pick on a very influential school of linguistic thought, a noun phrase may be anything: a single pronoun, a pronoun group, a single noun, a noun group, an adjective group, even a clause. Compare the two transformational phrase structure analyses in [21a] with our analyses in [21b]:

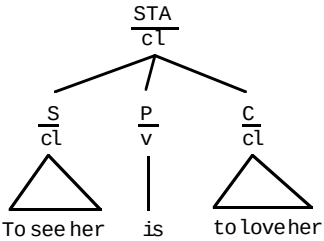
[21a]





[21b]





In the transformational analysis of *He left, He* is first analysed as an NP, which strictly speaking it is not; then, at a lower level of analysis, it is marked as a PRO. In our analysis, we do not beat about the bush, but stick to the simple fact: *He* is a pronoun with subject function. In the transformational analysis of *To see her is to love her, To see her* is also initially analysed as an NP, which strictly speaking, it is not. Then, at a lower level, its real nature is revealed: it is a clause; *to love her* receives a similar analysis. In our system, we say things more up front: *To see her* is a clause with subject function and *to love her* is a clause with complement function. The two transformational analyses provide far more of a pedagogical challenge than our analyses, to put it in positive terms. By assigning both function and (precise) form labels to constituents, our system captures more explicitly the distributional patterns merely implied by traditional phrase structure analysis.

F) Functions. Finally, maybe the most obvious difference between our system and the other systems that we have looked at is the inclusion of functions, the fact that we specify not only the form but also the function of all constituents in a sentence. Do we really need these functions? As indicated in my discussion of [21a] and [21b] above, the assignment of function labels gives us the freedom to be more precise in our assignment of form labels. But arguably this freedom comes at a high price. Chomsky has always argued against the assignment of function terms to constituents because functional values, he claims, can be deduced from purely formal properties and are therefore redundant (see e.g. Chomsky 1986:59ff). For example, any NP immediately dominated by S is a subject, any NP immediately dominated by VP is an object (cf. e.g. [14] above). However, this argument rests on the presence of the predicate VP as a valid constituent. In other words, Chomsky's approach may work – with the reservations noted – for his own phrase structures but not for the ones offered in our model. To avoid circularity, external evidence must be provided for the existence of the predicate. Even though such evidence is not too difficult to find, we hold that function terms are useful, and even essential in a pedagogical sentence analysis system. Without them we cannot establish a natural link between sentence analysis and the more specific grammatical rules governing the use of language. Imagine a student's grammar of English which does not refer to subjects, direct and indirect objects, subject and object complements, heads, dependents, etc.

Before I go on to discuss the problem of how to deal with predicates, let me briefly summarize the characteristic features of our system as presented so far:

Unlike other systems, our system:

[22] a) has no transformations or similar interstructural devices
b) does not operate with the predicate
c) allows multi-branching as well as binary branching
d) has no non-branching nodes except at word level
e) treats a group as a group, a clause as a clause, a word as a word
f) specifies both the form and the function of every constituent.

4. The pros and cons of operating with the predicate

There is a very long tradition for treating the subject and the predicate as the primary constituents of the sentence. This tradition probably goes all the way back to Plato's and Aristotle's 'ónoma' and 'rhêma', as Robins (1967:26ff) has pointed out. In more recent times, the subject and the predicate have been treated in more pragmatic terms as representing 'topic' and 'comment', respectively. Scholars rarely, if ever, question the legitimacy of operating with the predicate. Even Roman Jakobson seems to take its status as a universal category for granted. He says: "To instance simple relations among grammatical universals, we may cite the difference between the classes of nouns and verbs ... This difference is correlated but never merges with the likewise universal difference of two syntactic functions – subject and predicate" (1963:265).

Let us look more closely at some of the syntactic evidence for the existence of the predicate:

1) Coordination of predicates, as in *Jones [left his wife], [sold the house] and [moved to LA]*. Conventionally, if you can coordinate something, it must be a unit of some sort.

2) Fusion of a predicator verb and a following noun (sometimes referred to as object incorporation), as in

The incident *took place* before noon. *(happened)*
John *caught sight* of Bob. *(spotted, sighted)*
I *gave* her *a kiss.* (I *kissed* her)
I *took my leave*. (I *left*)

Such fusion is, if not predictable, at least more easily accounted for within a predicate framework. Constituents contain elements which are closely related in some sense. A close relation may develop into actual fusion.

3) AUX representation of predicates: certain pro-forms are capable of representing whole predicates:

> For years he wanted to *win the race* and he finally *did.*
> *Will* he *propose marriage to her?* Well, he *may.*
> They *are having a nice time in London. Are* they really!

In each of these examples, an auxiliary verb (*did*, *may* and *are*, respectively) represents a whole predicate. For example, *did* represents *won the race* or *did win the race.*

I think it is clear, on the basis of such evidence, that the predicate is not only an important pragmatic unit - the 'comment' - but also a syntactic unit which cannot be ignored. Nevertheless, I support the view that the predicate should not be treated as a regular sentence constituent in our sentence analysis. There are three reasons for this:

1) Although the predicate may have a reasonably clear function (i.e. that of 'comment'), there are no regular form types realizing the predicate. Unlike other functions it seems typically to be realized by combinations of form types, each serving more specific clause functions (i.e. predicator, object, adverbial, complement). This is why it is tempting to regard the predicate as a superfunction or 'metafunction'.

2) In those cases where the predicate is realized simply by a regular form type (a group or a word) as in intransitive constructions like *he left, she was laughing, they stopped*, etc., its presence creates unnecessary complexity by introducing a non-branching level of analysis. Nothing much is gained by analysing e.g. *was laughing* first as a predicate and then as a predicator.

3) There are other cases of closeness between constituents which must be taken seriously if we take the predicate seriously. The predicate is only one out of a number of possible metaconstituents. If we include one, why not the others?

This last point needs to be elaborated. Closeness between constituents is often a product of subordination or coordination. Here are

some examples of closeness between constituents in connection with subordination:

a) Dependent/head relations in noun groups with hypotactically related adjectives, as in:

[The [envious [Republican [senators]]]] complained

As mentioned earlier, in a sequence of adjectives, some adjectives may be more closely related to the head noun than others. In the example here, *Republican senators* is modified as a unit by *envious.* This phenomenon raises the question of what kind of constituent can be used to reflect different degrees of closeness between a head and its dependents in a noun group. Ideally we should have a metaconstituent to represent complex nominal heads. It is this kind of phenomenon that X-bar analysis is good at handling with its intermediate single-bar levels of structure.

b) Dependent/head relations in verb groups and predicates:

He [might [drop into a bar]]

The metaconstituent here *(drop into a bar)* is the *predication* (a predication is a predicate *minus* the operator, see e.g. Quirk et al. 1985: 79ff, 120ff). Note that there is strong evidence for treating the predication as a syntactic unit:

i. predications may be coordinated:

He [might [drop into a bar] and [down some liquor]]

ii. predications may be represented by PRO-forms:

I would like to go to Rome and *so* would my wife.

I gave Joyce a book and *so* did Bob.

iii. predications may be fronted:

Awaiting them were a tray of sandwiches, two bottles of wine, the director in uniform, and an exceptionally beautiful girl.

Deny it though he might, he dumped his wife in Paris.

Having offered some examples of closeness between constituents in connection with subordination, I now turn to constructions showing closeness between constituents in connection with coordination:

a) As we have already seen, constituents consisting of the predicator plus subsequent clause functions (i.e. the traditional predicate) may be coordinated:

Roger [left his wife], [sold the house] and [moved to LA]

The metaconstituent required for the analysis of such examples is the predicate.

b) Coordination of verb and preceding noun or noun group, as in

[John sold] and [Peter bought] the house.

Such examples point to the existence of a metaconstituent consisting of subject and predicator.

c) Coordination of units consisting of nouns or noun groups with different clause functions:

Jack gave [Belinda a kiss] and [Mary some good advice]
I told [Bob a white lie] but [Jenny the whole truth]

Such examples point to the existence of a metaconstituent consisting of indirect object and direct object.

d) Coordination of units consisting of noun or noun groups and preposition groups with different functions:

He sent [Jack to London] and [Jenny to Rome]

Such examples point to the existence of a metaconstituent consisting of a direct object and an adverbial.

e) Coordination of units consisting of nominal constituents and adjectival constituents with different functions:

He painted [the kitchen white] and [the bathroom yellow]

Such examples point to the existence of a metaconstituent consisting of a direct object and an object complement.

These examples by no means exhaust the possibilities.

5. Stacking

Having examined some of the problems in connection with the predicate, as well as other instances of closeness between constituents, we are now in a better position to formulate our requirements: what we need is a sentence analysis system which basically is as simple as

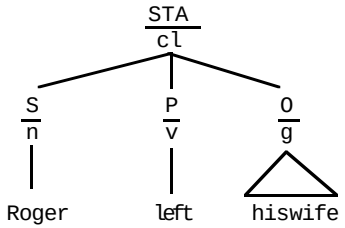
possible and as neutral as possible with respect to different interpretations of syntactic structure. At the same time, however, we need a special descriptive tool which allows us to show closeness relations between basic constituents.

Now, we already have a very simple and neutral system, namely the system proposed by Davenport, Dienhart, Larsen and myself. As we have seen, this system does not operate with the predicate, the predication or any other close relations between constituents. It merely offers basic, non-redundant, 'no-frills' analyses. As to the special tool required for the representation of close relations between constituents, I propose that we employ the 'stacking' technique developed by Davidsen-Nielsen and myself in connection with the current grammar project that we are undertaking for the Danish Research Council.

A stack is a collection of constituents somehow belonging together without obviously constituting one of the basic constituents of the sentence analysis system. It is a kind of syntactic 'wild card' – a characterization offered by my colleague John Dienhart – in the sense that it may represent any metaconstituent. There are form stacks and there are function stacks. A form stack (for which we use the symbol 'x') is a collection of forms constituting a derived, non-basic form type (i.e. a form type other than the four basic form types of the system mentioned in [2]: group, clause, compound unit and single word). A function stack (for which we use the symbol 'X') is a collection of functions constituting a derived, non-basic function (i.e. a function other than the functions already established for the system, such as subject, predicator, object, head, dependent etc.). By using a stack one shows a close relationship which cannot otherwise be captured by the system. The effect of introducing a stack into some constituent level in an analysis of a sentence is to *delay* the assignment of the basic forms and functions comprised by the stack to a level further down where this assignment is possible or desirable.

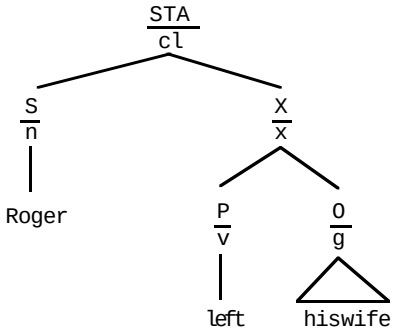
Let me offer an example. The basic analysis of a simple example like *Roger left his wife* looks like the analysis in [23] (note that we use the triangle to indicate that the internal analysis of a constituent is irrelevant for the argument):

[23]



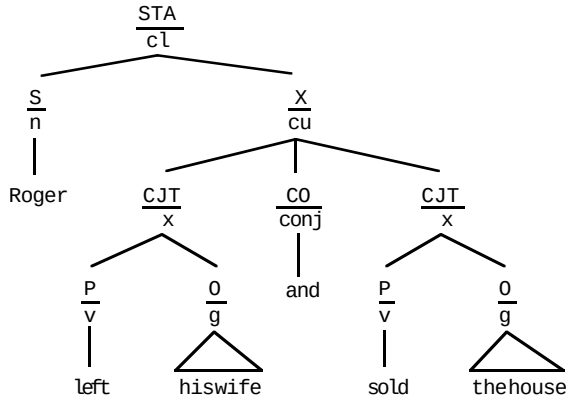
If, for one reason or another, we want to show the predicate in our analysis of this example (*left his wife*), it can be done in this way:

[24]



In our model, the predicate is not a basic function type and therefore has been treated as a function stack (X). Nor is the constituent *left his wif*e an established form type: it is not a group, a clause, a compound unit or a single word. We therefore have to treat it not only as a function stack but also as a form stack (x). In the case of the example *Roger left his wife*, we have a choice between the analysis in [23] and the one in [24]. Sometimes, however, we do not have a choice, e.g. when a sentence contains coordinated predicates as in [25] *Roger left his wife and sold the house*:

[25]



Any strictly surface structure analysis of this example requires the assignment of metaconstituent labels. Note that the function stack X and the form stack x are here separated: X represents a non-established function (coordinated predicates) but this function is realized by an established form type, the compound unit (cu). This cu consists of three established functions: two CJTs and a CO. The two CJTs are both realized by a non-established form type consisting of P and O (the traditional predicate). This is how stacking operates: by introducing stacks we delay the analysis of some material until we reach a level where we can handle the analysis in terms of our established metalanguage. Capital X is used when it is too early to assign function labels; small x is used when it is too early to assign form labels.

The concept of stacking allows us to offer more precise definitions of two major form types: groups and clauses. As a first approximation, we can say that a group consists of *at least* a head (H) and one dependent (DEP), while a clause consists of *at least* two of the following functions: S, P, Od, Oi, Cs, Co, A, SUB. For example, *the stranger* is a group consisting of the head *stranger* and the dependent *the*. In the sentence *To see her is to love her*, the subject *To see her* is a clause consisting of a P *(To see)* and an Od *(her)*.

To clarify what we mean by the terms 'group' and 'clause', it is necessary to supplement the minimal requirements ('at least a head and a dependent' and 'at least two clause functions') with the condition that

both groups and clauses are *maximal* forms: for a group to be a group we should not leave out relevant group constituents, and for a clause to be a clause we should not leave out relevant clause constituents. Let me illustrate this by referring once again to the analysis in [24] *(Roger left his wife)*. Why don't we simply treat the predicate *left his wife* as a clause rather than as a form stack. It complies with our minimal condition: it contains at least two clause functions, a P and an Od. The reason why we do not treat the predicate as a clause is that, in the particular example under analysis, it is only *part* of a clause. We have left out a relevant clause constituent, the subject *Roger.* Unlike clauses, the constituent *left his wife* cannot contain a subject of its own – there already is a subject outside the constituent (in other words, we cannot say e.g. **Roger [Roger left his wife*]). The clause being a maximal form type, we cannot simply leave out a relevant constituent and call *left his wife* a clause. Instead we call it a form stack, a metaconstituent containing closely related constituents.

Consider also an example like

[26] She was often visited by *the tall strangers* .

In this example we cannot analyse *tall strangers* as a head group within the group *the tall strangers*, even though *tall strangers* might function as a whole group in an example like *Tall strangers kept calling on Jenny*. The reason why *tall strangers* cannot be analysed as a head group within the italicized group in [26] is that in *that particular example* it is only part of a group. If we accepted it as a group, we would have to explain why it is incompatible with the definite article (since we cannot say **She was visited by the [the tall strangers]*). In [26] the DEP:art *the* is a relevant group constituent in relation to *strangers* or *tall strangers* and therefore should not be excluded from a group analysis of these units. If in our analysis we want to show that there is a close relationship between *tall* and *strangers* and that the definite article can be interpreted as relating to *tall strange*rs rather than just to *strangers*, this can be done by using stacking: *tall strangers* is on this interpretation an example of H:x.

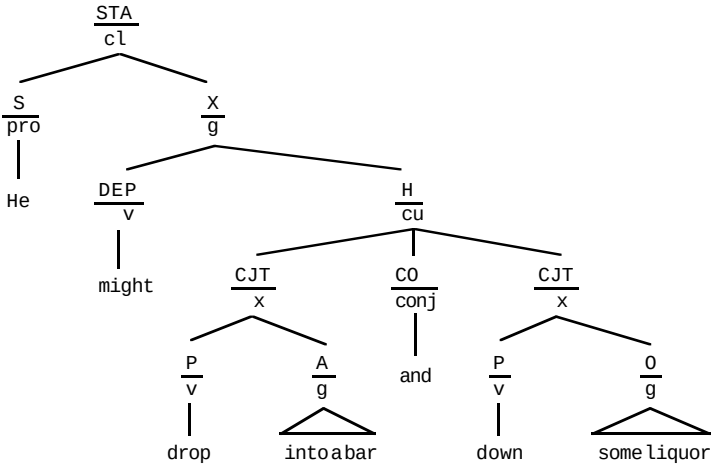
In other words, a group is the maximal form unit in any given case which can be analysed in terms of a head and one or more dependents. A clause is the maximal form unit in any given case which can be analysed in terms of two or more clause functions. Closeness relations

between group constituents within a group or between clause constituents within a clause are handled by the stacking technique.

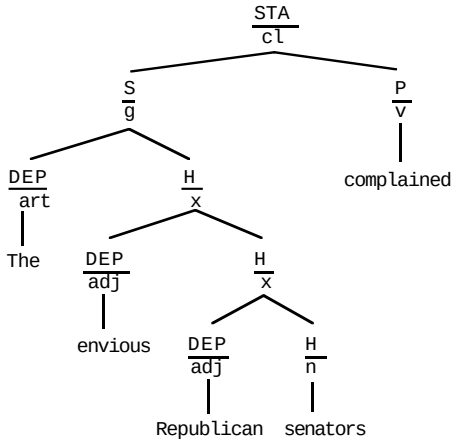
6. Stacking applied

Let me conclude this paper by applying the stacking technique to some of the other problem areas mentioned in section 4 above:

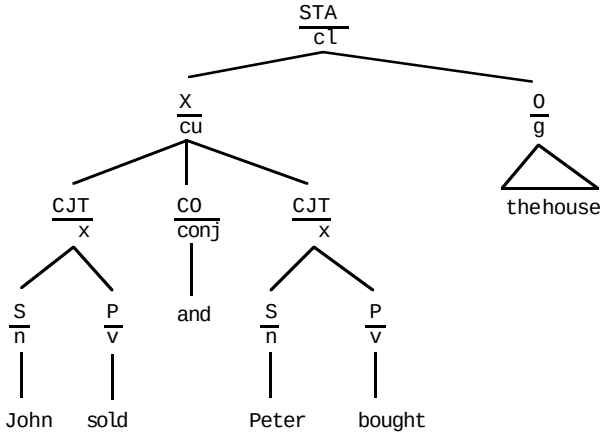
[27] Predication stacks:

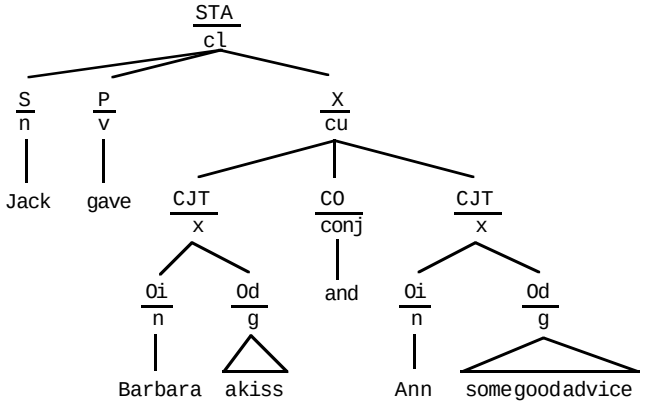


[28] Subordination within nominal groups:



[29] Coordination of stacks consisting of clause functions:





7. Concluding remarks

Let me finally touch on some of the alternatives to stacking. There are several possible ways of avoiding stacks. Within the school of generative syntax, many of the phenomena that I propose to handle by stacking could be described in terms of a strong transformational component. But recent developments within generative syntax tend to diminish the role of transformations.

Another solution might be to operate with strong filtering devices, i.e. with a reduction model with lots of missing constituents. Thus, an example like *He might drop into a bar and down some liquor* in [27] would be a reduced version of *He might drop into a bar and he might down some liquor*. The problem with this approach is that it seems odd to derive perfectly good sentences from more clumsy, unnatural sentences.

Yet another possibility is to increase the number of basic form and function terms in our metalanguage to cover all the different derived function and form stacks (such as e.g. predicates, S P collocations, etc.). There are two problems with such an approach: a) the system becomes far more complicated and difficult for students to learn; b) one loses the sense of similarity between different stacks. Different stacks are used for very similar purposes, mainly to show closeness between constituents which are not otherwise always close.

Finally, we might give up the condition that groups and clauses are maximal constituents. In a sense this is exactly what happens in X-bar analysis. Although we could get rid of form stacks in this way, the terms 'clause' and 'group' would be less clearly defined. And we would still need function stacks.

One argument against stacking is that it seems to be a rather *ad hoc* tool which we use whenever there is a need for it. There can be no denying that stacking has been invented in order to solve precisely the problems that our basic sentence analysis system cannot handle. And the proposed division of labour between the basic system and its extensions may well be arbitrary from a theoretical point of view. However, from a pedagogical point of view, it is sound to have stacking as an extension of the system rather than to try to change the basic system itself more radically.

Having said that, I believe that there is nothing ad hoc about my attempt to cope with the individual problems touched upon in this paper. In English, predicates, hypotaxis and all sorts of coordinated constituents are natural phenomena in need of adequate syntactic representation. We have to use *some* tool to cope with these phenomena, be it stacking or some other device. For the reasons stated, stacking is an attractive option.

References

Bache, C., M. Davenport, J. M. Dienhart & F. Larsen (1993): *An Introduction to English Sentence Analysis*, 2nd edition revised. Copenhagen: Munksgaard.

Bache, C. & N. Davidsen-Nielsen (1995): *Grammar Compendium*. Odense: English Department, Odense University.

Chomsky, N. (1986): *Knowledge of Language: Its Nature, Origin and Use*. New York: Praeger.

Davenport, M., J. M. Dienhart & F. Larsen (1984): "Computer Aided Instruction in English Sentence Analysis". *Bits and Bytes* 1, The English Department, Odense University.

Greenberg, J. H. (ed.) 1968 (1963): *Universals of Language*. Cambridge, Mass.: M.I.T.

Halliday, M. A. K. (1994): *An Introduction to Functional Grammar*, Second Edition. London, New York, Sydney & Auckland: Edward Arnold.

Hartvigson, H. H., J. M. Dienhart & L. Kvistgaard Jakobsen (1977): *Opbyggelige Øvelser i Engelsk Sætningsanalyse*. Odense: Odense Universitetsforlag.

Jacobs, R. A. & P. S. Rosenbaum (1968): *English Transformational Grammar*. New York: Wiley.

Jakobson, R. (1963): "Implications of Language Universals for Linguistics". In Greenberg, J. H. (ed.) *Universals of Language*, edition referred to: Cambridge, Mass.: M.I.T 1968.

Quirk, R., S. Greenbaum, G. Leech & J. Svartvik (1985): *A Comprehensive Grammar of the English Languag.*, London & New York: Longman.

Radford, A. (1981): *Transformational Syntax*. Cambridge: Cambridge University Press.

Robins, R. H. (1967): *A Short History of Linguistics*. London: Longmans.

APPENDIX

FUNCTIONS

S	Subject (= Real Subject)
Sp	Provisional Subject
Sr	Real Subject
P	Predicator
O	Object (= Real Direct Object)
Od	Direct Object
Oi	Indirect Object
Op	Provisional Direct Object
Or	Real Direct Object
A	Adverbial
C	Complement (= Subject Complement)
Cs	Subject Complement
Co	Object Complement
SUB	Subordinator
CO	Coordinator
CJT	Conjoint
H	Head
DEP	Dependent
STA	Statement
QUE	Question
COM	Command
EXC	Exclamation
PER	Performative
X	Function stack

forms

cl	clause
g	group
cu	compound unit
n	noun
v	verb
adj	adjective
adv	adverb
art	article
pro	pronoun
prep	preposition
conj	conjunction
infm	infinitive marker
num	numeral
intj	interjection
decl	declarative clause
inter	interrogative clause
imper	imperative clause
excl	exclamatory clause
Ø	zero
x	form stack
△	'unfinished analysis'