A Syntactic Description of German in a Formalism Designed for Machine Translation

Paul Schmidt
IAI-Eurotra-D
Martin-Luther-Str. 14
D-6600 Saarbrücken
West-Germany

Abstract:

This paper presents a syntactic description of a fragment of German that has been worked out within the machine translation project Eurotra. It represents the syntactic part of the German module of this multilingual translation system. The linguistic tool for the following analyses is the so-called CAT-framework.
In the first two sections of this paper an introduction of the formalism and a linguistic characterization of the framework is given. The CAT formalism as a whole is a theory of machine translation, the syntactic analysis part which is the subject of this paper is an LFG-like mapping of a constituent structure onto a functional structure.
A third section develops principles for a phrase structure and a functional structure for German and the mapping of phrase structure onto functional structure.
In a fourth section a treatment of unbounded movement phenomena is sketched. As the CAT-framework does not provide any global mechanisms I try to give a local treatment of this problem.

0. Introduction

There are two basic givens for Eurotra:

(i) Stratificational description of language.
The description of language consists of an analysis on three levels:
ECS (Eurotra-Constituent-Structure) which describes language according to part/whole relations and word order,
ERS (Eurotra-Relational-Structure) which describes language in terms of syntactic functions and
IS (Interface Structure) which describes language according to deep syntactic relations enriched by semantic information such as semantic features for characterizing lexical units.

(ii) The CAT-formalism.
The CAT-formalism is the linguistic tool for the description of language. As this formalism has no global mechanisms there are some restrictions concerning the treatment of unbounded dependencies.

Taking these givens into account, I would like to present the following topics:
(i) An introduction to the formal language as far as necessary for the treatment of the linguistic phenomena I would like to describe,
(ii) A characterization of the Eurotra stratificational description of language as a functionally oriented theory,
(iii) A development of principles for a syntactic description of German
(iv) A sketch of the treatment of unbounded dependencies

1. The formalism

I would like to introduce only those parts of the CAT-formalism which build the basis of my analyses. That is two kinds of rules:
(i) socalled b-rules (structure building rules). They build structures and transform structures into structures.
(ii) so-called feature-rules and killer-filters. They are put together into one class as both of them operate on structures created by b-rules expressing generalizations over attributes.

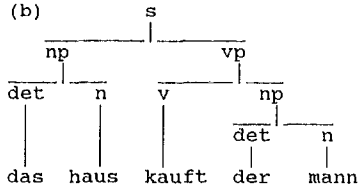
1.1. b-rules

(1)(a) {cat=s} [{cat=np},{cat=vp}].
(b) {cat=vp}[{cat=v},{cat=np}].
(c) {cat=np}[{cat=det},{cat=n}].
(d) {cat=v,lu=kaufen,lex=kauft,tns=tensed}

In (1)(a)-(d) we have b-rules, which define a small ECS-grammar. (d) is a rule for a terminal. The dominance relation is expressed by square brackets. The grammar in (1) assigns sentence (2)(a) structure (2)(b).

(2)(a) Das Haus kauft der Mann
(The house, the man buys)

(b)



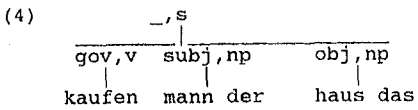
The same way as in (1)(a)-(d) an ECS-grammar was written we can write b-rules defining functional structures. (3) is a b-rule defining the functional structure for (2)(a):

(3)(a) {cat=s} [{sf=gov,cat=v,frame=subj_obj},
{sf=subj,cat=np,case=nom},
{sf=obj,cat=np,case=acc},
*** {sf=mod}].** (sf=syntactic function)

(b) {lu=kaufen,sf=gov,frame=subj_obj}.

b-rule (3) creates the functional structure (4) for sentence (2)(a).

(4)



The transformation will be done by the translation b-rule (5).

(5) ts1 = S:{cat=s} [NP1:{cat=np},
~:{cat=vp} [V:{cat=v},NP2:{cat=np}]]
=>
S:{cat=s} <V,NP2,NP1>.

A translation-b-rule (t-rule) consists of a left hand side (lhs) which defines a representation, in our case it would unify with the ECS-structure in (2)(b), and a right hand side (rhs) which defines a dominance and precedence relationship between the items represented by the variables (capitals). If there is a b-rule on the next level, in our case ERS, which satisfies these conditions, the translation succeeds. t-rule (5) says that structure (2)(b) shall be translated into a structure which is dominated by a node of category s which dominates the three items represented by the variables in the order given in the rhs of the t-rule. As the verbal governor, in our case 'kaufen', requires a subj_obj frame, expressed by the frame feature, (3)(a) is the ERS-b-rule which would match with the rhs of t-rule (5) and create (4).

1.2. f-rules and killer-filters

1.2.1. f-rules
f-rules and killer filters allow for the definition of a context part (those features after the slash) and an action part as example (6) shows. An f-rule applies to a representation only if the context part strictly unifies with the object.

(6) { case=C,nb=N,gend=G,/cat=np}
[{/cat=det},{case=C,nb=N,gend=G,/cat=n}]

(6) says that for each np consisting of a det and an n case, number and gender of the n have to be percolated into the mother node.
I would like to make two remarks: (i) the feature percolation in example (6) could be done in b-rules. Thus, it might seem that f-rules are superfluous. However, as section 4 will show, there are many cases where we need feature percolation by f-rules.
(ii) I will make a special use of f-rules. I will take everything as context and action part. That means, if f-rule f unifies with representation r, r will be replaced by the result of unification, if not, r survives unchanged.

1.2.2. killer-filters
Killer filters specify structures which are not well-formed and which therefore have to be deleted. We might imagine a rule which kills nps having a pronominal head and an np in genitive.

(7) killer-filter:
{cat=np} [{cat=det},{cat=n,n_type=pron},{cat=np,case=gen}].

2. CAT as a functionally oriented framework

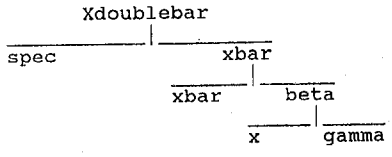
2.1. A comparison with a configurational framework
For the linguistic characterization of the Eurotra framework I would like to make a brief comparison between two kinds of linguistic theories:

(i) those which assume syntactic functions as universal primitives of language (prototypical: LFG)
(ii) those which claim that syntactic functions could be reduced to configurational facts (prototypical: GB).

Each of the two possible ways of describing language forces the linguist to describe linguistic facts as word order, binding relations, agreement, case assignment or long distance movement in a certain way.

The configurational framework claims that there is a general schema for phrase structure rules which is the universal pattern according to which all constituent structures of all possible languages are built. It is the x-bar schema:

(8)



(8) represents the x-bar schema, also D(eep)-Structure in GB. On this structure movement rules operate creating S(urface)-structures.
So, this is the kind of explanation a configurational framework gives: There is a canonical schema (the x-bar schema) and each configuration not fitting into this schema is explained as derived by universally restricted movement transformations.

The functional alternative has to rely on syntactic functions as universal primitives. So, phrase structure does not necessarily claim a universal status, and movement rules are not even necessary. This requires a different treatment of the linguistic phenomena. How does the CAT framework fit into this? The adoption of the three level system (ECS,ERS,IS) makes Eurotra functionally oriented as it adopts the way of linguistic description a functional approach has to adopt. While a configurational description consists in mapping given configurations onto a canonical schema, the x-bar schema, by explaining configurations which do not fit into x-bar as having undergone movement transformations, a functional description consists in a mapping of phrase structures onto functional structures.

2.2. Completeness and coherence in Eurotra
There is an essential which holds for all functional frameworks, namely the completeness and coherence principle.
This principle says: A functional structure is well-formed iff it is complete and coherent. A functional structure is complete iff it contains all the syntactic functions required by the frame of the framebearing element. A functional structure is coherent iff it contains only the required syntactic functions. Eurotra allows for the expression of this principle in two different ways:

(i) Enumeration of frames
The ERS grammar has to enumerate all possible patterns, all frames which are possible, as b-rules, and the value of the frame feature of the gov determines that only the wanted and nothing but the wanted governors go into the structure building rule.

(9) {cat=s} [{sf=gov,cat=v,frame=subj_obj},
{sf=subj,cat=np,case=nom},
{sf=obj,cat=np,case=acc},
***{sf=mod}]**

In (9) completeness is expressed by the fact that both framebound syntactic functions are obligatory. So, if one of the functions is missing, the structure is not well-formed. Coherence is expressed by the fact that the structure building rule only allows for the two syntactic functions and nothing else. This prevents e.g. the creation of an oblique object.

(ii) Completeness and coherence by f-rules and killers
There is, however another way of expressing completeness and coherence which does not require the enumeration of all frames. We need the following rules: (a) One ERS b-rule which enumerates all possible syntactic functions optionally as (10) does:

(10) :b: {cat=s} [{sf=gov,cat=v},
^ {sf=subj,cat=np,case=nom},
^ {sf=obj,cat=np,case=acc},
^ {sf=obj2,cat=np},
^ {sf=obl,cat=pp},
^ {sf=scomp,cat=s},
.....
***{sf=mod,cat=pp}**
^ {sf=topic}]

(b) A separate encoding of the functions a verb is subcategorized for, i.e. the frame feature is given up and a feature for each syntactic function is introduced:

(11) {lu=see,subj=yes,obj=yes,sf=gov}

All other syntactic function feature values will have to get the default "no" (by default f-rules).

(12) {lu=see,subj=yes,obj=yes,obj2=no,obl=no,scomp=no,sf=gov}

We can now state completeness and coherence independently by killer filters:

(13) :k: {cat=s} [{sf=gov,cat=v,subj=yes},
^ {sf=obj,cat=np,case=acc},
^ {sf=obj2,cat=np},
^ {sf=obl,cat=pp},
^ {sf=scomp,cat=s},
.....
***{sf=mod,cat=pp}**
^ {sf=topic}]

(13) determines that if the feature for subj=yes then there must be a syntactic function "subj" in this representation. Expressed by a killer it reads: if there is a structure whose gov has an a-v-pair subj=yes and contains only functions which are not subj then this structure is not well-formed. The same which has been stated here for subj can be stated for all functions. To get coherence we use a killer filter as in (14).

(14) :k: {cat=s} | {sf=gov,cat=v,subj=no},
{sf=subj,cat=np,case=nom},
*** {} |**

(14) says: If the structure whose gov has the feature-value 'no' for the feature 'subj' contains a feature bundle containing the feature sf=subj, plus anything else, then this structure is not well-formed.

3. Syntactic description of German

3.1. Principles of syntactic description

As we have seen above, the syntactic description of a language in Eurotra follows a functional approach. In our description this is not only reflected by the existence of a functional level but also by the nonhierarchical, nonconfigurational description of the sentence constituent we offer. As we do not use the given x-bar schema we need no empty elements on ECS and we describe German as a nonconfigurational language.

Though in German matrix clauses we have SVO word order, German is usually considered an SOV language. Matrix clause word order is considered as derived from subordinate clause word order by movement transformations. (cf. Koster 1975, Thiersch 1978, Reis 1985.

On this basis we would like to make another assumption concerning phrase structure which says that there is a unique structure underlying all German sentences (matrix clause and subordinate clause). This hypothesis is called "Symmetry Hypothesis" or "Doppelkopfanalyse" (cf. Reis 1985). It is shared by most of the generative syntacticiens such as H. den Besten, H. Haider, J. Lenerz and J. Koster. I will adopt some version of this "Symmetry Hypothesis" (SM) which will be developed in the following:

3.2. Phrase structure description (ECS) of German

(i) The initial base rule is (15)

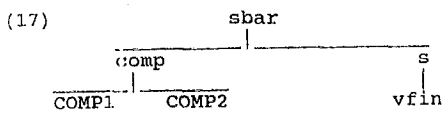
(15) sbar --> comp s

(ii) There are two left peripheral positions comp1 and comp2. We would like to represent this fact by the following expansion rule:

(16) comp --> COMP1 COMP2

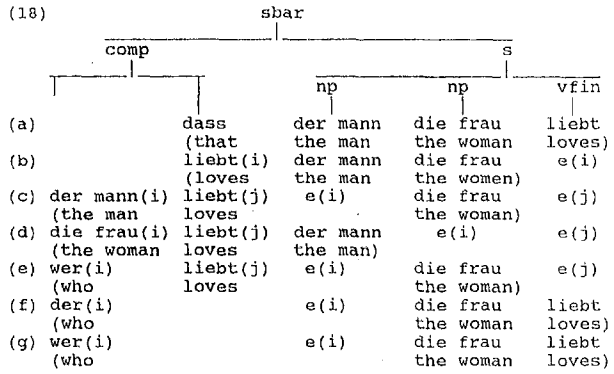
where COMP1 and COMP2 represent positions which will be described thus:

(iii) The B - position has the feature +- tnsd which specifies it as the verb/complementizer position, being filled in the base component only by lexical complementizers. This analysis yields the following structure:



(iv) Two movement rules operate on this structure, deriving all non SOV structures. These two rules are: T1 : Verb fronting and T2 : Topicalization where COMP2 is the landing site for the finite verb and COMP1 the landing site for X-double-bar.

We will show now in (18) how possible German sentence structures can be derived according to SM.

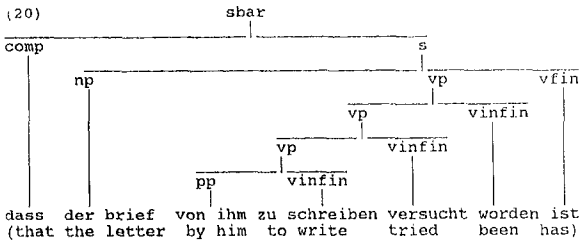


(18)(a) represents the base structure description. (18)(b) V/1 representation as in yes/no questions, the finite verb having moved into COMP2 leaving behind a trace. (18)(c) represents ordinary matrix clause word order derived by the two movement rules T1 and T2. (18)(d) represents matrix clause word order with a topicalized direct object. (18)(e) is a case of a matrix clause word order interrogative. (18)(f) a relative clause and (18)(g) a subordinate clause interrogative. (18)(e) and (g) represent a case of wh-movement. Untensed subordinate clauses which would not fit into this schema would be analysed as PPs:

(19) pp [p ohne [vp [v zu fragen]]]
(without asking)

This SH-analysis can at least make the following claims: (i) The COMP2-position as complementizer position and as landing site for the verb-fronting transformation nicely explains the relation between occurrence of complementizer and the occurrence of the finite verb (ii) As (18)(e) and (g) show, wh-movement can be represented equally for matrix clauses and subordinate clauses, namely as movement into COMP2. (iii) The SH-analysis is compatible with the productive traditional "Stellungsfelderhypothese" (cf. Olson 1984).

Another subject of phrase structure should be mentioned here: the treatment of the verbal complex. We adopted the following approach: Every verb is a full verb. Auxiliaries are subject control verbs (cf Netter1986, 1988, and Bresnan 1982).



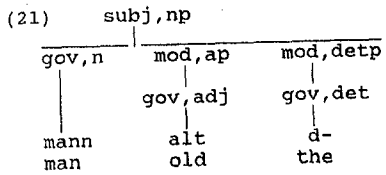
This treatment allows an easy calculation of tense, voice and aspect on ERS, as there is still structural information. As representation (20) shows, all nonfinite verbs are treated on ECS the same way, namely as the head of left recursively branching vp-constituents. This enables an easy treatment of auxiliaries as raising verbs on ERS (see section 3.3.).

3.3. Relational structure (ERS)

3.3.1. Principles

The relational structure of a language is constituted by the property of lexical units (lu) to bind certain other elements. This property is usually called "valency". Formally this fact is reflected in the formalism by the property of local trees. Each local tree contains just one gov(ernor), its valency-bound elements which are the comp(lements) and its non-valency-bound elements which are the mod(ifier)s:

(21)



The valency of a lu is its property to bind a certain number and a certain kind of syntagma. In other words: a valency theory is a theory on how many and which kind of syntagma occur given that a certain lu occurs. We consider verbs, nouns, adjectives and prepositions as having the property of being able to bind other syntagmas. A major part of every valency theory is the design of a test which is meant to determine the difference of complement and modifier. In the history of valency theory a lot of tests have been developed, among others the following: Elimination test, free addability test, adverbial clause test, verb substitution test, do-so-test, backformation test. We adopted a revised addability test (cf. Schmidt 1986).

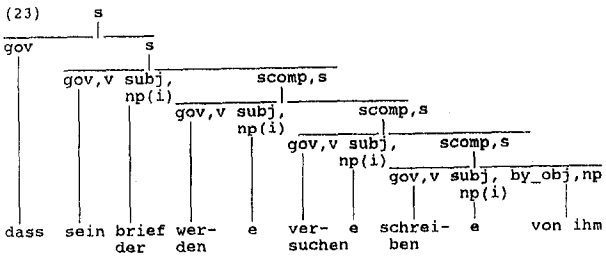
3.3.2. Word order
The most important aspect with the decription of the relation between ECS and ERS is that the present formalism allows for the treatment of free word order languages. We consider German as having a relatively free word order. The decisive feature is that the rhs of the t-rules are able to specify only dominance relations which is expressed by the parenthesis in (22). Permutations in the German middle field can easily be treated as shown in example (22).

(22) S:{cat=sbar}
[~:{cat=comp} {TOPIC,V:{cat=v}],
~:{cat=s}[ADV1:* {cat=adv2},
NP1:{cat=np},
ADV2:* {cat=adv2},
NP2:{cat=np},
ADV3:* {cat=adv2},
NP3:{cat=np},
ADV4:* {cat=adv2},
VP:{cat=vp}]]

=>
S:<(TOPIC,V,ADV1,NP1,ADV2,NP2,ADV3,NP3,ADV4,VP)>

3.3.3. The verbal complex on ERS
As shown in structure (20), auxiliaries are analysed as full verbs. The structural analysis in (20) makes it easy to treat auxiliaries as raising verbs on ERS, as (23) shows.

(23)



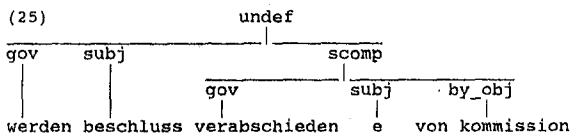
3.3.4. Passive
The problem with passive is the following: There is a relation between the two sentences in (24)

(24) Die Kommission verabschiedet den Beschluss
(The Commission adopts the decision)
Der Beschluss wurde von der Kommission verabschiedet
(The decision was adopted by the council)

which is in terms of surface syntactic frames that the phrase being the subject in (a), namely 'die Kommission', is the by_obj in (b) and the direct object of (a) is the surface syntactic subject of (b) (bearing all features surface syntactic subjects usually have, as e.g. nominative case). In terms of thematic roles we could say that the agent is in both cases 'die Kommission' once realized as an NP in nominative case, once realized as a PP with the preposition 'von'.

We keep surface syntactic information and aim at the following structure:

(25)



In universal grammar passive usually is treated in a general way, as passivization is considered a universal process:
- In GB passivization is considered as a movement process which is contained in the general move alpha schema.
- GPSG also treats passive on the syntactic level in form of a metarule.
- LFG being a "lexicalist" theory treats passive in the lexicon by a lexical rule which is s.th. like

(subj) -> zero/(by obj)
(obj) -> subj

In Eurotra we have neither of these devices, neither movement rules nor metarules, nor lexical rules. However, it seems as if we could simulate the lexical rules just by putting the "active frame" into the b-rule as in (26).

(26) {cat=s,voice=passive} | {sf=gov,cat=v,frame=subj_obj},
{sf=subj,cat=np,case=nom},
{sf=by_obj,cat=pp,pform=von),
*** {}]**
{lu=verabschieden,sf=gov,cat=v,frame=subj_obj}

This has the same effect as the LFG lexical rule: only one encoding of the verb with its active sub_obj-frame is necessary.

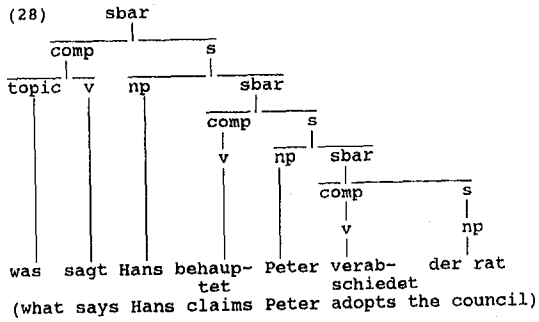
4. Treatment of Unbounded Movement Phenomena

4.1. Wh-movement
4.1.1. The Representation
I would like to explain my approach with an example:

(27)(a) was sagt Hans, behauptet Peter, verabschiedet der Rat
(what says Hans claims Peter adopts the council)
what does Hans say that Peter claims the council adopts
(b) was sagt Hans

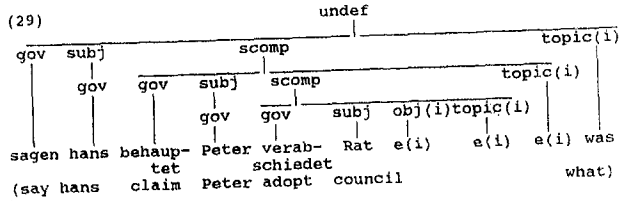
According to our ECS grammar the following ECS tree is created:

(28)



We imagine a functional representation like (29).

(29)



In (29) we can see that a chain was created from the topic of the

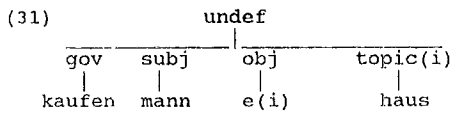
matrix clause via the topic of the embedded clause to the correct syntactic function slot. We have to guarantee that it is a correct chain which I understand as a chain that is correctly coindexed with the correct function in the ERS b-rule.

4.1.2. The Creation of the Correct Structure

The structure in (29) is created by inserting empty elements by t-rule application in a very controlled way. I would like to give an exemplification by NP-complements. Structure insertion by t-rules exploit the fact that movement has its landing site which is the node called compl in representation (17). In the lhs of the t-rule this information is exploited. We also know that each phrase which occupies the compl position on ECS has to go to an ERS slot which has sf=topic. We need the four t-rules for doing the job.

```
(30) ts1= S:{cat=sbar}
        [~:{cat=comp,tns=tensed}
             [TOPIC:{cat=np},V:{cat=v}],
         ~:  {cat=s,tns=untensed}
             [NP2:{cat=np},~:^{cat=punct},SBAR:^{cat=sbar}]]
 =>
 S:{cat=s}
    <V,NP2,{cat=np,n_type=empty},SBAR,TOPIC:{sf=topic} >.
```

The t-rule in (30) treats local wh-movement as in (2)(a) and creates structure (31).



(30) creates an empty np-slot which has to be interpreted as one of the b-rule slots subj, obj or obj2 in (10). It will go to sf=subj,sf=obj and sf=obj2. It is up to completeness and coherence to determine that (31) is well-formed in our case.

For the top of an unbounded dependency construction (29), we need t-rule (32) which puts the topicalized np into the topic slot on ERS, but without creating a corresponding empty np.

```
(32) ts2= S:{cat=sbar}
          [~:{cat=comp,tns=tensed}
               [TOPIC: {cat=np}, V: {cat=v}],
           ~:{cat=s,tns=untensed}
               [NP2: ^{cat=np},
                    ~: ^{cat=punct},
                  SBAR: ^{cat=sbar}]]
=>
S:{cat=s} < V, NP2, SBAR, TOPIC:{sf=topic,cat=np} >.
```

(33) treats the middle of unbounded dependency constructions i.e. a sentence structure which has an empty topic. The middle builds the link between embedded sentences and matrix clause . It has no empty correspondent in the structure. This structure is created by a t-rule which operates on an ECS representation which has an empty topic landing site (see (28)).

```
(33) ts3= S:{cat=sbar} [~: {cat=comp,tns=tensed} [V:{cat=v}],
                    ~:  {cat=s,tns=untensed}
                        [NP2: ^{cat=np},
                             ~: ^{cat=punct},
                           SBAR: ^{cat=sbar}]]
=> S:{cat=s} < V,NP2,SBAR,{cat=np,n_type=empty,sf=topic} >.
```

For the bottom of the structural representation we finally need a t-rule which creates an empty topic and an empty corresponding np. (34) is this rule. It is also applied only under the condition that the ECS landing site for wh-movement is empty.

```
(34) ts4= S:{cat=sbar} [~:{cat=comp,tns=tensed} [V:{cat=v}],
                    ~:{cat=s,tns=untensed}
                        [ NP2: ^{cat=np},
                             ~: ^{cat=punct},
                              SBAR: ^{cat=sbar}]]
=>
S:{cat=s}
       <V,NP2,{cat=np,n_type=empty},
                SBAR,{cat=np,n_type=empty,sf=topic } >.
```

We now have all the pieces needed for creating the correct structures which can occur in unbounded dependency structures. (28) only represents a three-fold s-structure, however rule (33) caters for all possible middles as it will be applied as many times as there are middles.

A few comments seem to be in order on these rules: (30) and (32) on the one hand and (33) and (34) on the other hand have the same lhs which might cause overgeneration.

Rule (31) caters for the case that the s is the matrix-clause containing a moved NP which has to find its functional slot downwards somewhere in a functional structure of an embedded clause. For this case we need a topic which has no correspondent on the same level.

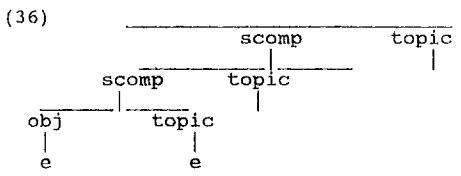
If we take (27)(b), rule (30) as well as rule (32) will be applied, both of them putting "was" into the topic function, (30) creating an empty NP-slot, (32) not creating an empty NP. So, we have two rules (30) and (32) which apply to the same lhs producing two different ERS structures. The completeness and coherence test determines which t-rule (30) or (32) creates the correct structure. Both of them will be applied but only one, namely (30) creates the correct structure according to the completeness and coherence criterion. In the case of rule (33) and (34) we have the same problem. Both of them apply to the same lhs, once inserting an empty np, once not. Again, completeness and coherence has to determine whether the result of (33) or (34) is correct.

4.1.3. Feature Checking

The creation of the correct structure is only half of the story. We have not guaranteed yet that only correct structures are created and above all that only correct chains are created. This will be done by an interaction of f-rules percolating the relevant features such as gender, number, case and the index feature and by killer filters which guarantee that only correctly indexed chains survive. First of all we need f-rules which percolate the relevant features.

```
(35) :f: a_top_to_s = {cat=s}
        [ {sf=gov},
          ^{sf=subj},
          ^{sf=obj},
          ^{sf=obj2},
          {sf=scomp,top_index=I,top_nb=N,top_gend=G},
          *{sf=mod},
          {sf=topic,index=I,nb=N,gend=G} ].
```

(35) is an example which percolates number, gender and index from topic to scomp. Another f-rule of the same style will percolate these features from scomp to the topic node of the embedded sentence, and a third f-rule from topic to the empty functional slot. So, if we consider example (28) the percolation of the relevant features follows the following path:

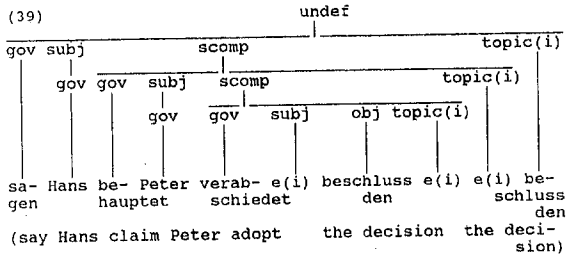


The same kind of f-rule will percolate the case feature independently the same path. (For the reasons see below). For feature checking we need killer rules which kill all structures which are not correctly indexed and those which represent an empty chain. E.g. (37) is a rule which deletes all structures where the case feature of the empty topic and the corresponding empty np is not the same.

(37) :k: ktopic3= {cat=s} [{sf=gov,cat=v},
***{},**
{cat=np,type=empty,case~=C,index=I},
***{},**
{sf=topic,cat=np,case=C,index=I}].

Actually we need another 6 killers which check number and gender.
Rule (37) makes clear what has been the sense of the separate case-feature-percolation. If we percolated the case feature in rules like (35) we could not use the index - feature for feature checking. I would like to explain this with an example. We need a rule to filter out the wrong representation (39) which is the representation of the following ill-formed sentence:

(38) * Den Beschluss sagt Hans, behauptet Peter, verabschiedet den Beschluss
* (The decision says Hans, that Peter claims, adopts the decision)

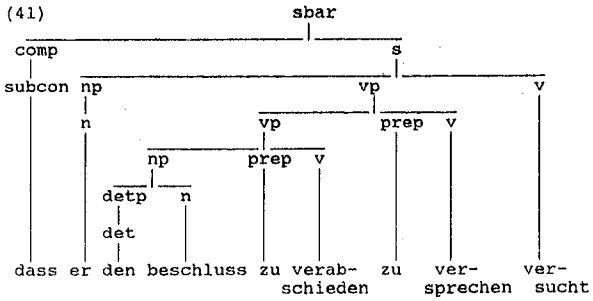


According to our f-rules the index is percolated down into the empty subject slot in the lowest scomp. (It cannot go elsewhere). This subject has case=nom which is stated in the ERS b-rule. The case feature is the means to get rid of the wrong chain as there will be a clash between the "arriving" case=accusative and the already stated case=nominative. If the case feature had not been percolated independently we would not have any possibility of applying killer rule (37) as the f-rule would not have been applied for the reason of the impossibility of unification. My rules percolate the index into the subj-slot and make possible the application of (37).

4.2. Control
Let us consider the following case of subject control:

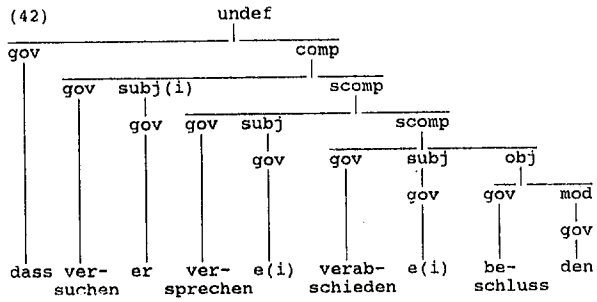
(40) dass er den Beschluss zu verabschieden zu versprechen versucht
that he tries to promise to adopt the decision

Our ECS-grammar would assign the ECS-structure (41):



The ERS representation would look like (42).

In the case of control-structures it is easy to control the insertion of structure by t-rule as embedded control structures are vps in our system. As we have seen in section 3, each vp is lacking a subject np which is inserted on ERS by t-rule (43):



(43) tvp1 = VP:{cat=vp}
[NP1:{cat=np},VP:{cat=vp},~:{cat=prep},
V:{cat=v,tns=untensed}]
=>
VP:{cat=s} < V,{cat=np,type=empty,sf=subj},NP1,VP >.

In control structures feature checking works the same way as in wh-constructions. We only need a correct feature percolation which puts the relevant features to the scomp-node and from there to the subj-slot. We only have to take care that in the scomp-node features are not confused with topic-features. This can be guaranteed by using ctl_case etc. in scomp.

(44) :f: f_ctl1 = {cat=s} [{sf=gov,cat=v,ctl=subj},
{sf=subj,cat=np,nb=N,gend=G,index=I},
***{},**
{sf=scomp,cat=s,ctl_nb=N,ctl_gend=G,ctl_index=I},
***{}].**

5. Summary

The descriptions of a significant fragment of German above seem to be a good basis for a translation system. The functional structures created in our system can easily be mapped onto deep syntactic predicate-argument-structures which are enriched by semantic information. From there transfer should happen.
As far as the treatment of unbounded dependencies is concerned there might be some problems in transfer. Certain pied piping phenomena and multiple wh-movement might make necessary a more powerful mechanism.

6. Literature:

Abraham,W.(ed)(1985) Erklärende Syntax des Deutschen, Tübingen, (=Studien zur deutschen Grammatik 25).

Arnold,D. et al.(1987) The Eurotra Reference Manual, Release 2.1., ms. Utrecht.

Bresnan,J.(1982) The Passive in Lexical Thoery, in: Bresnan,J The Mental Representation of Grammaticasl Relations Cambridge, Mass./London Engl.

Koster,J.(1975) Dutch as an SOV Language. Linguistic Analysis 1,pp.111-136.

Lenerz,J.(1984) Diachronic Syntax: Verb Position and COMP in German, in: Toman (1984).

Netter,K.(1986) Getting Things out of Order. An LFG Proposal for the Treatment of German Word Order, Coling Proceedings (1986),p 494 - 496.

Olson,S.(1984) On Deriving V-1 and V-2 Structures in German, in: Toman (1984).

Reis,M.(1985) Satzeinleitende Strukturen. Ueber COMP, Haupt- und Nebensaetze, w- Bewegung und Doppelkopfanalyse, in: Abraham (1985).

Steiner,E., Schmidt, P., Zelinsky, C. (1988) (forthcoming) from Syntax to Semantics. (New Insights from Machine Translation). London 1988.

Schmidt,P.(1986) Valency Theory in a Stratificational MT System,in: Coling Proceedings (1986).

Thiersch, C.: Topics in German Syntax, unpub. Diss. 1978.