SALAT:
Machine Translation Via Semantic Representation

Christa Hauenschild, Edgar Huckert, and Robert Maier

Universität Heidelberg, Angewandte Sprachwissenschaft,
Landfriedstr. 12, D-6900 Heidelberg, Fed. Rep. of Germany

This paper is to show how some semantic problems are handled in the machine translation system SALAT (= System for Automatic Language Analysis and Translation)[1]. Therefore it will be necessary to give a rather comprehensive introduction into the system. Otherwise the topic could not be treated in a satisfactory way, and it would be impossible for the reader to get a sufficiently detailed and concrete idea of the involved problems and proposed solutions.

1. Basic Principles of SALAT

In the development of SALAT the following principles have been and are being respected:

- theoretical orientation of the project;
- independence of individual languages;
- strict separation of data from algorithms;
- translation via a logico-semantic interlingua representing meanings of translated sentences and knowledge about the situation and the world ("Situations- und Weltwissen").

1.1 Theoretical Orientation

The project of SALAT aims at theoretical rather than at practically applicable results, i.e. its main orientation is not towards problems of large-scale application (e.g. optimizing of programs, preparation and processing of comprehensive data sets), but towards the investigation of the prerequisites for fully automatic high-quality translation.

By this term coined by Y. BAR-HILLEL in the early sixties we understand (as he did e.g. in (1) and (2)) machine translation without pre- or post-editing respecting the usual standards of human translation. We agree with Y. BAR-HILLEL that such a translation must be based on a full-fledged semantic analysis, i.e. on the 'understanding', of the translated text by the computer, which is possible only if "good general background knowledge" (p. 331 in (1), "knowledge about the world" in our terms) as well as supplementary information about the situation of utterance is available; for otherwise e.g. the resolution of many ambiguities would be impossible. But we do not agree that therefore fully automatic high-quality translation is a priori infeasible nor that the only way is to confine oneself to machineaided translation or to utterly restricted domains, as was claimed by Y. BAR-HILLEL in (1) and (2).

Rather do we hold that it still makes sense to try and approximate high-quality machine translation by working out concepts for the necessary components of an appropriate translation system, including a data base (containing knowledge about the situation and the world) and an inference-making device (for a realistic data base would have to be very large, but nevertheless could not comprise all the information needed for any special translation problem), and by defining the corresponding algorithms in a way which allows them to be programmed and tested on gradually enlarged sets of examples. The main difficulty in designing such a system lies in the absence of an entirely satisfactory theory - not only of knowledge representation, but of the translation process as a whole - that would make it possible to formulate problems and solutions in a way precise and explicit enough for computational processing.

Thus such a project can be promising only if it is paralleled by the development of efficient theoretical approaches for its different parts and based on further results of linguistic, logical and artificial-intelligence research. Then it may contribute to a general theory of translation and serve as an appropriate tool for the verification of the adequacy of the different sub-theories, linguistic or other. So the theoretically oriented approach to machine translation makes sense, even if it turned out that fully automatic high-quality translation was in fact impossible (which would be very difficult to prove, as the underlying theories can surely always be improved), on account of the lots of useful insights gained on the way as to its precise limits and the linguistic or other reasons therefor.

1.2 Independence of Individual Languages and Separation of Data from Algorithms

In accordance with the theoretical orientation of SALAT, the translation procedure is not dependent on special source or target languages: it is being tested on German, English, French, and Russian data, which is documented in (6) (esp. Vol.II), (12), (13), (14), (15), (17).

As identical formalisms and algorithms ought to be applicable for the translation between different language pairs, one of the main programming principles consists in a strict separation of data (e.g. lexica, rules) from algorithms (e.g. application of rules). This is convenient, too, for the division of labour between the linguist and the programmer. Moreover, the rule systems can be revised and completed without changing the programs (the reverse is only partially true; mostly, however, a revision of programs does not require substantial, but just formal changes of rules, if any), which is of great importance in the testing phase of a machine translation system.

1.3 Logico-Semantic Interlingua

The translation process goes via a logico-semantic interlingua which serves as a general representation system for the meanings of translated sentences as well as for those of sentences containing supplementary information about the situation and the world, possibly not inferable from the translated text, but needed for high-quality translation. As the interlingua representations of these latter sentences are to be stored in a data base and operated on by special deduction systems, the interlingua must be flexible enough to represent natural-language meanings in a way that facilitates the formulation of logico-semantic deduction rules.

The interlingua of SALAT is the core of the whole system, as it were, for it serves as a basis for the solution of semantic translation problems. Its form is described in some more detail in 2.1, while its functioning is explained and illustrated in several passages in chapters 3 to 5.

2. Formalisms Used in SALAT

For the different levels of natural-language description and for the connections between them the following formalisms have been developed or adapted for SALAT :

- ε-λ-context-free syntax for "deep structures" (= expressions of the interlingua);
- quasi-normal two-step context-free syntax in complex notation for "surface structures";
- transformational rules establishing relations between deep and surface structures or among deep structures.

The rôles of these formalisms in the different parts of SALAT are explained more fully in chapters 3 and 4.

It ought to be kept in view that, while the formalisms as such seem to be satisfactory, this does not hold for the present versions of the respective rule systems, which are rather limited in scope and precision, but can nevertheless serve as a basis for realistic tests of the system. Moreover, the flexibility of the formalisms facilitates the adaptation of analyses of natural-language phenomena formalized differently. So the rule systems can easily be revised and completed, in order to cover gradually enlarged fragments of natural languages and of the translation relations between them.

In the following paragraphs, the formalisms used in SALAT will not be described at full length, but rather illustrated with the aid of hopefully illuminating examples.

2.1 ε-λ-Context-Free Syntax

The interlingua of SALAT is an ε-λ-context-free language designed by A. v. STECHOW (21) with reference to ideas by R. MONTAGUE (23), P. SUPPES (22), M. J. CRESSWELL (7), and U. EGLI (9), which was modified by us according to the special needs of our project (19), (20): a context-free syntax, the terminals of which are "semantemes" mainly representing lexical meanings of natural languages, is enlarged by rules introducing variables and the abstractors ε and λ for classes and functions respectively (in the present version only λ is used). The deep structures, i.e. expressions of the interlingua, generated by this "deep syntax" can be understood as representations of type-theoretic expressions in the form of trees and therefore have nearly nothing in common with Chomskyan deep structures. They allow a model-theoretic interpretation which is used in the heuristics of the deduction rules, for the control of their correctness, and for the formulation of meta-theoretical statements.

```
U------------------------------------------------------------
|                                                           |
|                                                           |
UQ------------------------------------------------------    ZEIT
|                                                     |     |
|                                                     |     |
S-------------------------------------------------    SIT   $Z1
|                                               |     |
|                                               |     |
N------                                         C1    $S1
|     |                                         |
|     |                                         |
Q     C1------------------------------------    $BELL
|     |                                   |
|     |                                   |
$DEF  MC1----                             C1
      |     |                             |
      |     |                             |
      KC1   C1-----                       $HUND
      |     |     |
      |     |     |
      $UND  X'    S------
                  |     |
                  |     |
                  N     C1-----
                  |     |     |
                  |     |     |
                  X     C2    N------
                        |     |     |
                        |     |     |
                        $HAB  Q     C1
                              |     |
                              |     |
                              $IND  $HERR
```

<u>Fig.1</u>
Sample deep structure[2]

The deep structure in Fig.1 represents the German sentence "Der Hund, der einen Herrn hat, bellt.", which translates into "The dog which has a master barks."[3]. The non-terminal node labels, i.e. "deep categories", to be found in the example correspond approximately to the following constituents (mainly of surface syntax):

U - utterance
ZEIT - time of utterance
UQ - utterance without time specification
SIT - situation of utterance
S - sentence, clause
N - noun phrase, proper name
Q - quantifier, determiner
C1 - "one-place contentive" = noun, one-place verb or verb phrase, intersective adjective, ...
C2 - "two-place contentive" = two-place verb or verb phrase, ...
MC1 - modifier of one-place contentive
KC1 - connective for one-place contentives

The terminal labels of the deep structure are variables ("X"es, the indices of which are contained in a special list omitted here) and semantemes (marked by "$") corresponding to German lexemes with the exception of "$Z1" and "$S1", which are arbitrary names of the time and the situation of utterance respectively. The other semantemes may be related to the following English lexemes:

$DEF - the
$UND - and
$HAB - have
$IND - a
$HERR - master
$HUND - dog
$BELL - bark

The embedded "S" of Fig.1 shows the typical deep structure of a relative clause. The rule introducing this "S" originally contained "λ" (C1 → λ X_i S), which is left out here just for technical reasons (binarity of the structure), but is of course considered in the model-theoretic interpretation. Interpreting the "sentential part" of the sample deep structure, i.e. without the categories referring to the utterance and its specification, would yield the following result:
$BELL ιx($HUNDx & ∃y($HERRy & $HAByx))

"$HAByx" is to be read as "x has y", which does not correspond to the usual order of arguments in the predicate calculus. The full sequence of the steps of interpretation is given in (14), pp.14-15.

2.2 Two-Step Context-Free Syntax in Complex Notation

Surface structures of input or output sentences in SALAT are expressions of a context-free language defined by a quasi-normal two-step context-free syntax in complex notation, where "two-step" refers to the fact that it consists of a "word syntax" (roughly: morphology) and a "sentence syntax" (for the details of the formalism and the corresponding analysis procedure see (4), more or less comprehensive rule systems written in this formalism for fragments of different languages are contained in Vol.II of (6) and in (15)).

The following figure shows the context-free surface structure of the English sentence "The dog which has a master barks.", i.e. of the sentence corresponding to the German expression represented in the deep structure of Fig.1. This surface structure was the output of a sample translation with SALAT described in (14), pp.29-63.

```
SENT----------------------------------------------
1                                                |
|                                                |
NOPH----                                         PRED----
2      |                                         3      |
|      |                                         |      |
ARTI   NOUN----                                  PRST   PREN
4      5      |                                  6      7
|      |      |                                  |      |
THE    NOUN   RECL----                           BARK   S
8      9      10     |                           11     12
       |      |      |
       NOST   REPR   PRED-----------
       13     14     15            |
       |      |      |             |
       DOG    WHICH  PRED----      NOPH----
       16     17     18     |      19     |
                     |      |      |      |
                     PRST   PREN   ARTI   NOUN
                     20     21     22     23
                     |      |      |      |
                     HA     S      A      NOST
                     24     25     26     27
                                          |
                                          MASTER
                                          28
```

Fig.2

Sample surface structure

The meanings of the syntactic categories in Fig.2 should be to some extent self-explanatory by virtue of the structure assigned and the abbreviations employed. The boundary between word syntax and sentence syntax is defined by the "transition categories" (here: "ARTI" - article, "NOUN" - noun, "REPR" - relative pronoun, "PRED" - verb). The labels of the nodes dominated by them belong to the word syntax (here: "NOST" - noun stem, "PRST" - verb stem, "PREN" - verb ending).

The digits under the node labels are the canonical node numbers, which could not be omitted here (as in Fig.1) because of their function in the following list of "subcategories", which is an integral part of the surface structure.

Figure 3 contains the "syntactic subcategories", which are related to their respective "main categories" (i.e. non-terminal labels of the surface structure) by the canonical node numbers of the latter, listed in the left column. Together they form "complex categories"; so this is just a special notation for complex node labels which keeps the tree structures within sensible dimensions[4]. The "specifications" of the subcategories (comparable to values assigned to variables) are given in parentheses behind the subcategory names.

node number	subcategories
2	NUM(1)
3	ERG(1)NUM(1)PER(3)
4	NUM(1)
5	NUM(1)ART(2)
6	ERG(1)PTY(2)
7	NUM(1)PER(3)PTY(2)
9	NUM(1)ART(2)
10	NUM(1)ART(2)
13	ART(2)NTY(1)
14	ART(2)CAS(1)
15	ERG(1)NUM(1)PER(3)
18	ERG(2)NUM(1)PER(3)
19	NUM(1)
20	ERG(2)PTY(1)
21	NUM(1)PER(3)PTY(1)
22	NUM(1)
23	NUM(1)ART(1)
27	ART(1)NTY(1)

Fig.3 Subcategory list

Subcategories are the core of the complex notation of syntactic rules (cf. (4), pp.56-57); e.g. they are responsible for number and person agreement of subjects and predicates, for number (and gender) agreement within noun phrases etc. The range of possible functions of subcategories (even for more 'semantic' phenomena) is very wide; they are employed not only in the context-free syntax, but also in the transformational rules of SALAT (cf. 2.3).

The names of the subcategories occurring in Fig.3 are partly self-explanatory ("NUM", "PER", "CAS"); the rest shall not be explained in detail here (it is done in (14), pp.49-50). Only some remarks showing, hopefully, the flexibility of the subcategory device: "ERG" is responsible for verb valency, "PTY" and "NTY" specify the inflectional types of the stems and the corresponding endings of verbs ("PTY") and nouns ("NTY"). "ART" indicates the types of nouns with respect to potential relative clauses modifying them, namely whether they are to contain "who" or "which" ("ART(1)", indicating the "who" type, and "ART(2)", indicating the "which" type, are assigned to the nouns "master" and "dog" respectively, cf. nodes 23 and 9). The corresponding specifications characterize the relative pronouns (here: node 14 with complex label "REPR ART(2)CAS(1)", dominating "WHICH") and are 'inherited' from them by the relative clauses (here: node 10 with complex label "RECL NUM(1)ART(2)"), so that an 'agreement' of "ART" specification between the noun and the relative clause can be stated in the corresponding syntactic rule of noun modification[9].

2.3 Transformational Rules

Transformational rules are applied in SALAT for the transduction of surface structures into deep structures (second part of analysis) and vice versa (synthesis) as well as for deductions and other transformations in disambiguation and transfer. Like the SALAT deep structures, they exhibit little similarity to the Chomskyan concepts having the same name; they are just mappings of (sequences of) trees on (sequences of) trees. They are of a very flexible type conceived with reference to ideas by U. EGLI (9). In particular, they allow the formulation of supplementary conditions of application using decidable predicates with structures and parts of structures as their arguments. These predicates must hold true for the "object trees", i.e. the trees where the transformations are to operate, or else the respective rules will not be applicable (for a more comprehensive description see (14) and (16)).

The following figure shows a very simple sample rule with a decidable predicate occurring in it. The rule is given in tree notation, which is easier to read than the SALAT input notation (cf. Fig.5), especially in the case of more complicated structures. Moreover, the subcategories are combined with their main categories to build up complex node labels.

```
$BELL   --->   ?PRED
               ERG(1)
               |
               |
               ?BARK

Condition:
DOMS(S(N*C1),1)
```

Fig.4 Sample rule 1 - tree notation

The sample rule is taken from the translation with SALAT described in (14), namely from the "lexicon of semantemes and lexemes" (cf. chapter 3) of the English synthesis. It contains a semanteme on its left-hand side and a "lexeme" and an "intermediate category" on its right-hand side. Lexemes (in the SALAT sense of the word) appear only in the course of synthesis as "intermediate" labels between the semantemes and the terminal labels of surface structures. They belong to the intermediate categories (intermediate between deep and surface categories), which are all prefixed with "?", in order to distinguish them from the surface categories, from which they are different by virtue of their usually incomplete subcategory string.

The rule in Fig.4 will apply, if there is a terminal node labelled "$BELL" to be found in the object tree and if the sequence of its predecessors (in the ascending direction) starts with any number of "C1" nodes followed by one "S" node[6]. This condition is coded in the two-place predicate "DOMS", the first argument of which is an expression defining the properties to be fulfilled by the sequence of predecessors (here: one "S" dominating n times "C1") of the node referred to by the second argument. The latter is the canonical number of the respective node in the "rule tree" of the left-hand side of the transformational rule (here: "1" refers to the one and only node of the left-hand side).

This supplementary condition, called "structural restriction", could not be expressed easily in the left-hand rule tree on account of the indefinite number of "C1" nodes that may occur in the sequence of predecessors of "$BELL".

There are other decidable predicates defined in SALAT with different types of arguments. They can be combined to more complex structural restrictions with the aid of the operators of propositional logic (cf. (16), chapter 3.2.1); in particular, they can be negated. E.g. "CONTA" is frequently used with negation, claiming that a special substructure is <u>not contained</u> in a specified subpart of the object tree. It is not obvious, how this condition might be expressed in the rule tree without using a predicate (for an example of "CONTA" see the deduction rules in 5.2).

If the rule in Fig.4 is applicable, the terminal node of the left-hand side (i.e. the semanteme "$BELL") is substituted in the object tree by the terminal structure of the right-hand side (i.e. the lexeme "?BARK", dominated by "?PRED ERG(1)").

```
<$BELL>.
BED.:<DOMS(S(N*C1)1)>.
--->
<1.?PRED(?BARK)>.
1.ERG(1)
```

<u>Fig.5</u> Sample rule 1 - SALAT notation

In the notation used for transformational rules in SALAT both sides of the rule, as well as the structural restriction, are given in the form of bracketed expressions corresponding to the rule trees in Fig.5 and to a bracketed Polish notation of the structural restriction ("BED" is short for "Bedingung", i.e. "condition"). Here, the subcategories are related to their respective main categories by reference numbers ("1.", "2." etc.), which is very convenient in cases where different main categories and/or different occurrences of one main category have the same sequence of subcategories: they then get the same reference number, and the respective subcategory string has to be written down only once (cf. Fig.6).

In the next figure we give another example of a transformational rule to be applied in the English synthesis. This rule, too, stems from the sample translation in (14). It describes the substitution of the deep category "N" by the intermediate category "?NOPH" and the completion of the subcategory string of "?NOUN".

Another facet of the flexibility of the transformational formalism employed in SALAT is shown here, namely the use of variables of different types. Again, we have chosen a very simple, but nevertheless realistic rule, in order not to introduce too much complication that would only obscure the principles to be illustrated (for more complex rules see 5.2 and the rule systems in (12), (13), (14), (17)).

```
N--------         --->  ?NOPH
|        |              NUM(A)---
|        |              |       |
?ARTI    ?NOUN          |       |
NUM(A)   ART(B)         ?ARTI   ?NOUN
|        |              NUM(A)  NUM(A)
|        |              |       ART(B)
ΔA       ¤B             |       |
                        ΔA      |
                                ¤B

<N(1.?ARTI(ΔA)2.?NOUN(¤B))>.
--->
<1.?NOPH(1.?ARTI(ΔA)3.?NOUN(¤B))>.
1.NUM(A)
2.ART(B)
3.NUM(A)ART(B)
```

Fig. 6 Sample rule 2 - tree notation and SALAT notation

The prefixed "A"s and "B"s occurring in Fig.6 as terminal labels of the rule trees are variables ranging over substructures of the object tree to which the rule is applied. As identical values have to be assigned to different occurrences of one variable on both sides of the rule (as well as, by the way, in the structural restriction), the corresponding substructures are left unchanged in the course of the transformation, but they need not be specified, so that the rule can apply in different environments.

The different prefixes of the substructure variables serve for the distinction between different subtypes of them: "Δ" variables, "¤" variables, and "◊" variables correspond to exactly one edge in the object tree, to a non-empty sequence of edges, and to a (possibly empty) sequence of edges respectively. Such a distinction is convenient for the sake of the efficiency of the algorithm.

The "A"s and "B"s that appear in Fig.6 in "NUM(A)" and "ART(B)" are variables ranging over subcategory specifications. They are needed in the complex notation of transformational rules: the sample rule comprises, as it were, 4 elementary rules with constant specifications, for the occurring subcategories can both be specified in two ways, namely "(1)" or "(2)" (cf. the explanations to Fig.3), and all the combinations of specifications are permitted. In the case of more complicated rules with numerous subcategories and a great number of possible specifications, this leads to an important economy of rules and augments the transparency of the rule systems.

It is impossible to give a complete survey of the different possibilities of using variables in the transformational rules of SALAT in the space available for this paper. There are other devices, that were not at all mentioned here, e.g. the description of several subparts of the object tree in one transformational rule or the generalization of the formalism to many-place rules which operate on a sequence of object trees. The latter is exemplified in the deduction rules in 5.2 (for further details see [16]).

3. Major Parts of the Translation Process

The translation procedure of SALAT comprises three main programs, COSY (context-free syntax), TRANSFO (transformations), and DEDUKT (deductions, with different versions for disambiguation and transfer), which are employed in the following major parts of the translation process:

- two-step context-free analysis (COSY),
- transformational analysis (TRANSFO),
- disambiguation (TRANSFO and DEDUKT I),
- transfer (TRANSFO and DEDUKT II),
- transformational synthesis in two phases (TRANSFO).

3.1 Flow Chart of SALAT

The flow chart in Fig.7 illustrates the interconnections of the main subprocedures of SALAT with their input and output and with the data used in them.

The left column contains the input and output of the whole translation procedure as well as the intermediate results produced in the course of a translation. The central column shows the succession of the main parts without indication of their internal modular structure. In the right column, there appear the more or less permanent data used for several translations. They include rule systems for the description of the source language (SL), of the target language (TL), and of the relations between them as well as deduction rules and supplementary information coded in the data base in the form of deep structures.

The data base is the only component of the right column that will have to be partially changed in the course of one translation, for e. g. information contained in the input sentences will have to be added to the data base, in order to allow inferences referring to the context. Ideally, some parts of the data base and the rest of the data used are absolutely permanent, but in the testing phase of a translation system they will have to be gradually revised and completed.

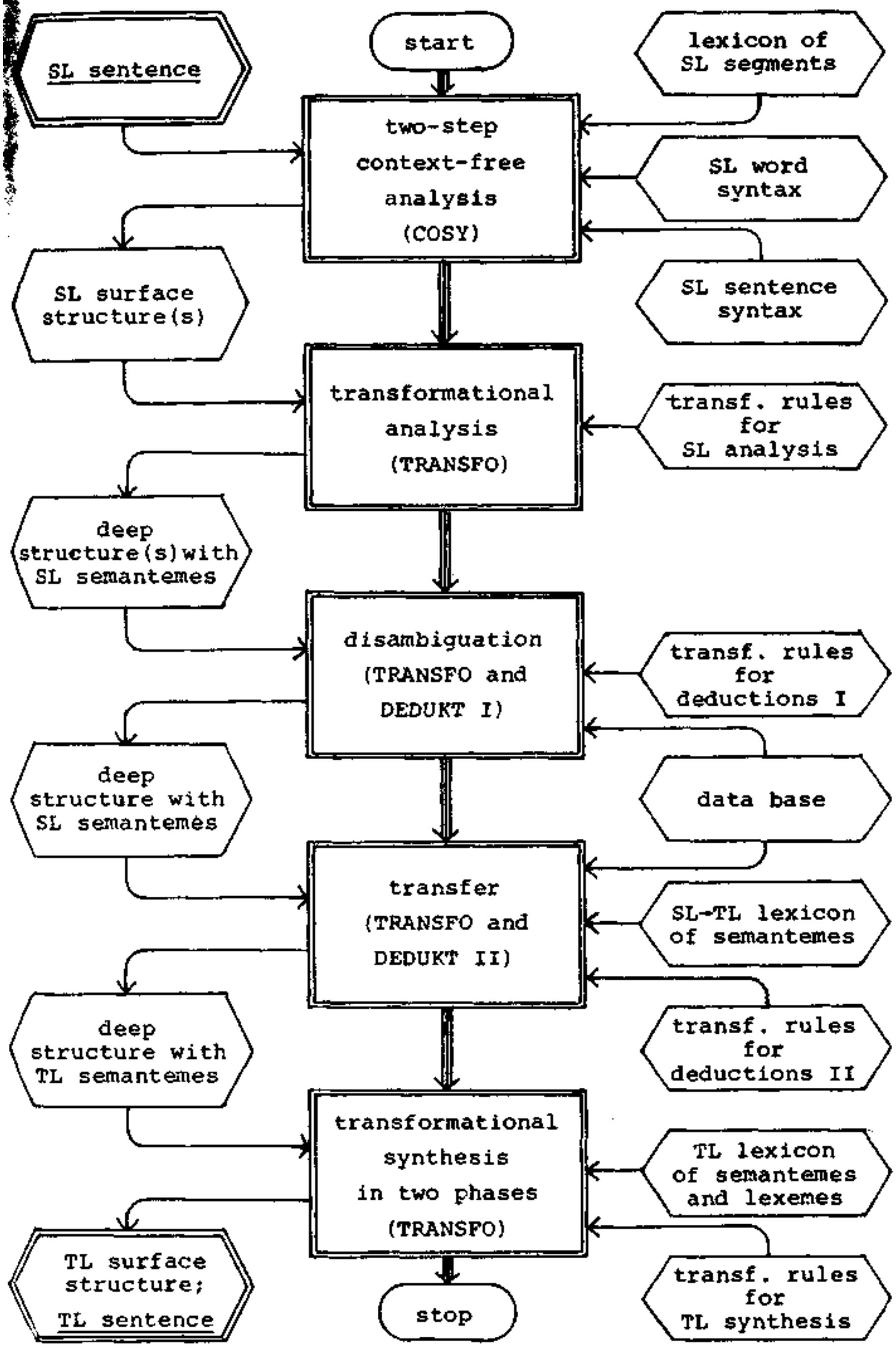

Fig.7 Flow chart of SALAT

3.2 Analysis and Synthesis

In describing the different parts of the translation process of SALAT, we do not proceed chronologically, but rather treat together analysis and synthesis. These have much in common, being the more 'syntactic' components compared to the essentially semantic steps of disambiguation and transfer, which are dealt with in 3.3.

By "analysis" we understand the transduction of source-language sentences into expressions of the interlingua, where several deep structures correspond to one sentence, if it has more than one reading. For reasons of efficiency the first part of analysis is accomplished by the context-free parser COSY, described in (5), which applies a two-step context-free syntax as illustrated in 2.2. This does not imply a general limitation of accepted languages to context-free ones, for the first part of the analysis may be skipped for non-context-free phenomena (if such exist?) or else the context-free syntax may accept some non-grammatical sentences which are then eliminated by the filtering power of the transformational part of the analysis.

The output of the context-free analysis is one or, in case of syntactic or lexical ambiguity (not e.g. in case of ambiguity of pronominal reference, which is dealt with in the transformational part of analysis), several surface structure(s) assigned to the input sentence.

The second, transformational, part of the analysis transduces surface structures into deep structures, thereby augmenting - if necessary - the number of different structures for one input sentence. The transformational rules of this step are applied by the procedure TRANSFO, described in (16), which constitutes an algorithm generating all the derivations rendered possible on the basis of the input sets of object trees and of transformational rules.

TRANSFO does not, by itself, guarantee the termination of the derivation process, for it accepts e.g. also cyclically operating rules which may be applied an infinite number of times. This is avoided by the special form of the rule systems used in SALAT: they exhibit an intrinsic rule ordering that is based on the utilization of mutually disjunct partial vocabularies for node labels, related to different states of the translation process. We think that an <u>ex</u>trinsic rule ordering is not necessary, for all the phenomena mentioned e.g. in (18) can surely be treated without it, if some fixed ideas on the possible and permissible forms of deep structures and transformational rules are given up. For more details regarding the rule ordering in SALAT see (11), (14), and (16); for a short discussion of the decidability problem connected with the use of transformational rule systems cf. (11), chapter 3.

TRANSFO is applied in synthesis, too; for "synthesis" means the transformation of deep structures into context-free surface structures of the target language the terminal strings of which are the wanted target-language sentences. It seems advantageous to guarantee well-defined (e.g. by a context-free syntax) output structures for theoretical and practical reasons (the Chomskyan surface structures are not well-defined in the sense intended here).

On account of its intrinsic rule ordering, the transformational synthesis is divided into two phases:

- a bottom-up phase where semantemes are substituted by target-language lexemes with the aid of a set of transformational rules called "lexicon of semantemes and lexemes", which simultaneously attaches incomplete morpho-syntactic information in the form of syntactic subcategories to the preterminal nodes (see sample rule 1 in 2.3); this information is then transferred to the higher nodes and partially completed by other rules (see sample rule 2 in 2.3);
- a top-down phase where complete morpho-syntactic information is attached to every node, so that finally the lexemes at the terminal nodes can be substituted by inflected word forms or by "segments" (i.e. by stems and endings forming together inflected word forms).

In any of the two phases, structural changes can be carried out if necessary.

3.3 Disambiguation and Transfer

In SALAT the synthesis cannot start directly with the deep structures resulting from the analysis. This is mainly due to the ambiguities occurring in the translation process. There are at least two kinds of them, namely ambiguities of the source language without reference to any target language and ambiguities of the source language in relation to a special target language. Let's call them "internal" and "translation" ambiguities respectively.

Internal ambiguities are e.g. such of quantifier scope, of pronominal reference, of homonyms; translation ambiguities occur e.g. if a lexeme which is not ambiguous from the point of view of the source language requires different translations into the target language in different contexts and/or situations of utterance. This holds for instance for the translation of English "put" into German "stellen, legen, ...", depending on properties of the involved objects. Internal ambiguities ought to be dealt with independently of the target language, whereas translation ambiguities should be resolved in a phase where the dependence of the target language is explicit.

However, it seems impossible to define a clear-cut boundary between the two kinds of ambiguities in practice. There are many uncontroversial cases, but numerous debatable ones, too. Nevertheless, there are good reasons to draw a provisional line separating them. It would be contrary to any principle of economy (and, probably, even impossible) to resolve all the potential translation ambiguities independently of the target language in question, at least in a machine translation system which, in principle, operates on any source or target language. A great amount of disambiguation work, not needed for the target language at hand, but only for some other one(s), would have to be carried out.

On the other hand, it does not seem satisfactory either to resolve all the ambiguities (including the internal ones) in a phase where the target language already comes in. In our system, this would imply that a meaning representation had to be assigned to input sentences where even ambiguity of homonyms and of pronominal reference were not resolved, which would be rather counter-intuitive. Moreover, as the resolution of ambiguities is one of the core problems (may-be even the one core problem) of machine translation, it is convenient to split up the task into different sub-tasks, in order to reduce its complexity.

Accordingly, the two kinds of ambiguities are resolved in SALAT in two different steps of the translation process: "disambiguation" and "transfer" deal with the internal and translation ambiguities respectively. Both components operate on the level of deep structures, which corresponds to the essentially semantic character of the involved problems.

Internal ambiguities are reflected in the following way: as the analysis may result in several deep structures representing the different readings of the input sentence, a disambiguation procedure is needed to find the reading(s) favoured by the context and/or situation, taking into consideration general knowledge of the world, too. So these decisions are based on information stored in the data base or inferable from it by logical deductions. These are transformations like those used for analysis and synthesis, but they are controlled in a different way by the procedure DEDUKT.

Translation ambiguities are mirrored in SALAT in non-correspondences between the semantemes of the deep structures resulting from analysis or disambiguation, which are oriented towards the source language, (short: "SL semantemes") and the lexemes of the target language. So, before the lexicon of semantemes and lexemes (i.e. the first step of synthesis) can be applied, the SL semantemes have to be substituted by TL semantemes. This is accomplished by a set of transformational rules called "lexicon of semantemes". These rules contain special conditions of application referring to information stored in or deducible from the data base, in order to make the right choice among the different potential translation equivalents (represented by different TL semantemes). The procedure deciding whether these conditions are met is again a version of DEDUKT which may in some respects differ from the version for disambiguation.

While the algorithms constituted by COSY and TRANSFO for analysis and synthesis are by now available in a form which is next to final (they are implemented and tested for data sets of different languages), this is not true for the algorithms of disambiguation and transfer, which are only partially completed. A first version of DEDUKT II (transfer deductions) is illustrated in 4.3 and 5.3.

4. Solving Semantic Problems in the Transfer Step

The semantic task of resolving translation ambiguities in the transfer step of SALAT is described in some more detail in this chapter and exemplified in chapter 5. Since the realization and partially even the conception of transfer is not yet finished, only a first rather incomplete version with quite a few obvious shortcomings is implemented and tested by now (cf. chapter 3.6.1 of [17]). Nevertheless, it may serve for an illustration of the main principles and methods conceived so far for the treatment of some semantic problems in SALAT. Moreover, by running rather simple examples one can gain useful insights into the involved problems and possible solutions, which is a fertile basis for further developments.

The components of the present transfer version in SALAT are the following:

- lexicon of semantemes, containing transformational "transfer rules";
- data base;
- deduction rules, controlled by DEDUKT II.

4.1 Lexicon of Semantemes

In SALAT no analysis into "semantic primitives" (as e.g. in [24], pp.123ff.), "atomic semantemes" or anything similar is performed. Rather there is one semanteme of the interlingua corresponding to each internally unambiguous lexeme vs. to each reading of an internally ambiguous lexeme of every involved language. It is possible, but will not occur very often, that one and the same semanteme is assigned to (the readings of) different lexemes of one language or even of different languages, namely if they are synonymous in a strict sense of the word*. Thus the sets of SL semantemes and of TL semantemes (corresponding to (readings of) SL and TL lexemes respectively) may overlap.

Usually, if the semanteme in question does not belong to the intersection of these two sets, there will be different potential translation equivalents (represented by different TL semantemes) the choice among which is to be made with regard to the context and/or situation, including general knowledge of the world. For instance, in translating English "put" into German "stellen" or "legen" (this does, of course, not exhaust the possibilities of translating "put") in SALAT, the semanteme "$PUT" has to be substituted by one of the semantemes "$STELL" or "$LEG". Under normal conditions, "$PUT" has to be transferred into "$STELL", if the involved object usually rather stands than lies, and into "$LEG", if it rather takes a lying position. Obviously this is just a first approximation; realistic descriptions of semanteme correspondences (representing lexeme relations) will have to consider much more complex information and a great number of different cases.

The substitution of SL semantemes by TL semantemes, including the choice among the latter, is performed in SALAT by the lexicon of semantemes. It is constituted by transformational rules the left-hand sides of which apply each to a single SL semanteme (in a certain structural context), while their right-hand sides contain TL semantemes or structures with only TL semantemes as terminal labels. If there are n possibilities of translating an SL semanteme, this yields at least n rules with this semanteme occurring on their left-hand sides.

The different cases are distinguished in the structural restrictions of the transfer rules with the aid of a special decidable predicate "DAB" (short for "data base"). "DAB" does not refer to structural properties of the object tree, as structural restrictions usually do (cf. 2.3), but claims that the information represented by its argument (e.g. the information about the preferred position of the object in the "put" example) is to be found in or deduced from the data base by a suitable deduction procedure. This means that the arguments of "DAB" have to be deep structures, for the data base contains information only in the form of such. Since, moreover, the transfer is accomplished by transformational rules like those for analysis and synthesis, no new formalism is necessary for their formulation.

The transfer rules for a certain combination of source and target language constitute the lexicon of semantemes for this language pair.

There exist not only 'lexical', but also 'structural' translation ambiguities. In the present version, no special rules of 'structural transfer' are provided for, because structural changes can easily be carried out in the course of synthesis. Such rules might as well be included into the lexicon of semantemes in a form which would not refer to single SL semantemes only, but to bigger substructures of the SL-oriented object trees.

4.2 Data Base

The data base of SALAT must provide all the information needed for the decisions to be made in disambiguation and transfer. This does not mean, however, that all the necessary information has to be explicitly contained (in the form of deep structures), which is in fact impossible, but it must be logically deducible from what is explicitly stored in the data base.

The relevant information cannot be contained in the deep structures alone due to the fact that their meanings are determined by the rules of model-theoretic interpretation only relative to an interpretation function for the semantemes. Such a function, however, does not exist explicitly anywhere in the implemented system (which would not be possible), but the interlingua expressions in the data base determine the meanings of semantemes - and thus the meanings of deep structures - in a quasi-axiomatic way (e.g. in the form of meaning postulates). So the data base replaces, as it were, an interpretation function for semantemes.

As can be concluded from diverse remarks on the function of the data base in several preceding paragraphs, there are different kinds of information, exhibiting different degrees of generality, to be stored in the data base:

- contextual information (from the input sentences),
- information about the situation and time of utterance,
- general knowledge of the world,
- meaning postulates.

Since there is no clear-cut boundary to be defined between these types of information, it is convenient to express all of them in one and the same formalism, namely in the ε-λ-context-free syntax of the SALAT interlingua. This is possible, because the meaning of any natural-language sentence as well as any statement on situations and times of utterance, on facts about the world, and on meaning relations in the involved languages can, in principle, be represented by interlingua expressions, i.e. deep structures:

- the meaning of a natural-language sentence is, of course, represented by the deep structure that is or would be assigned to it by analysis and disambiguation;
- statements on situations and times of utterance are expressed by deep structures containing their names;
- general facts about the world can be expressed in natural-language sentences, so they are represented accordingly;
- meaning relations are reflected in deep structures marked as valid at all times and in all situations (examples of such meaning postulates are given in 5.2).

As was already pointed out in 3.1, the data base contains (ideally) constant as well as changing parts. Meaning postulates and general knowledge of the world are constant; situational and contextual information change, usually even in the course of the translation of one text.

4.3 DEDUKT (II)

For each decidable predicate occurring in the structural restrictions of transformational rules in SALAT, a decision procedure must be available within TRANSFO. For the predicate "DAB", used in transfer rules, this procedure is DEDUKT II: TRANSFO calls DEDUKT II for the decision whether the argument of "DAB" can be found in or deduced from the data base by logical deductions (in the following paragraphs we shall often simply speak of "DEDUKT" without reference to its different versions, for most of what is said is true of DEDUKT I, too).

The logical deduction rules in SALAT are mappings of (sequences of) deep structures - namely of data-base expressions or intermediate results of deduction - on (sequences of) deep structures - namely on intermediate results or, finally, the arguments of "DAB". Thus these rules are just like those used in other transformational components of SALAT with the following exception: in general, they apply to more than one input structure and yield a corresponding number of output structures. This generalization to many-place rules is provided in the formalism of transformational rules in SALAT (cf. 2.3); so no new formalism for deduction rules is needed. Moreover, since DEDUKT applies transformational rules, several subprocedures of TRANSFO can be used for it, though there are substantial divergencies between the two procedures.

It is advantageous, too, that the logical deductions operate on deep structures. Since deep structures can be viewed as type-theoretic expressions in tree notation, the deduction rules of type theory can serve as an orienting basis for the formulation of the deduction rules of SALAT. Besides, correctness proofs for the latter are available on account of the model-theoretic interpretation of the deep structures.

As the set of all the logical consequences from the data base is not decidable, DEDUKT cannot offer a decision procedure for this set precisely. In artificial-intelligence research, theorem provers are often used in similar cases (see e.g. (8)). Theorem provers are complete deduction algorithms deducing every logical consequence from a given set of sentences. Nevertheless, they do not constitute a decision procedure, for, if a sentence is not deduced after a certain number of steps, it is impossible to decide whether it is deducible at all. Moreover, theorem provers operate in rather an undirected way, producing lots of irrelevant results. Because of time limitation, only an arbitrary subset of the possible consequences is generally drawn in practice, often excluding the most interesting ones and even rather easily decidable cases.

Therefore, DEDUKT should not be based on a theorem-prover concept. It is not necessary to make all the inferences, but rather to find the useful and relevant ones in a reasonable amount of time, although this means that sometimes only an ad-hoc decision, if any, is attainable. This does still not exclude the possibility of fully automatic high-quality translation, for more than once even human translators have to content themselves with ad-hoc solutions of translation problems, if a perfect solution cannot be found in the time and with the information available. Besides, human translators obviously do not use complete decision procedures.

So, what is needed for DEDUKT, is a deduction strategy oriented towards the relevant inferences. Such a strategy must lead to a step-by-step procedure where each step corresponds to the application of an appropriate set of deduction rules in a suitable order to a suitably chosen subset of the data-base expressions and where any step is carried out only if no solution has been found in the preceding one.

As no comprehensive and satisfying set of heuristic principles for such a strategy is known by now, only some simple heuristic advices have been respected in the implementation of the first version of the transfer step:

- split up the contextual information into simple pieces and use it by preference*;
- apply deduction rules to data-base structures only if the

latter have semantemes in common with the sentence (more precisely: the structure) to be proved;
- specify general statements with regard to the actual parameters, e.g. by applying universally valid sentences to the actual situation of utterance.

Obviously, these principles can be improved in several ways and extended in different directions. It will be useful, for instance, to conceive special decision procedures for easily decidable cases or to classify the data-base expressions according to their relative import (e.g. information coded as semantic features in other approaches may be particularly important). Since no strategy will be optimal for every case, different strategies ought to be employed regarding e.g. the form of the structure to be proved.

5. Transfer Example

In this chapter we give an example from German-French translation which comprises the following elements:
- deep structure of the translated sentence,
- transfer rules,
- data-base structures,
- deduction rules,
- steps of deduction.

5.1 Sample Sentence and Transfer Rules

The sentence to be translated is "Das alpine Gebiet ist klein." ("The Alpine region is small."), the deep structure assigned to it (with a somewhat simplified representation of "klein"[10]) is given in Fig.8 (for the meaning of the deep categories see the explanation to Fig.1).

```
U------------------------------------------
|                                         |
|                                         |
UQ----------------------------------      ZEIT
|                                   |     |
|                                   |     |
S------------------------------     SIT   $Z1
|                           |       |
|                           |       |
N-------                    C1      $S1
|      |                    |
|      |                    |
Q      C1--------------     $KLEIN
|      |             |
|      |             |
$DEF   MC1-----      C1
       |      |      |
       |      |      |
       KC1    C1     $GEBIET
       |      |
       |      |
       $UND   $ALPIN
```

Fig.8 Deep structure of the sample sentence

The translation of this sentence into French ought to yield: "La rêgion alpine est petite.". Ignoring other possible difficulties involved in this example, we focus our attention on the translation of German "Gebiet" into French. Out of the different potential translation equivalents we consider only "rêgion" and "domaine" (corresponding approximately to English "region" and "domain").

The relevant rules of the German-French lexicon of semantemes can be stated informally as follows:
1. Substitute the "Gebiet" semanteme by the "rêgion" semanteme, if the "Gebiet" is a geographic one.
2. Substitute the "Gebiet" semanteme by the "domaine" semanteme, if the "Gebiet" is a scientific one.

This is, of course, only a first approximation of realistic transfer rules, which is indeed even desirable for the sake of the transparency of the example.

In the linear SALAT notation these rules read:

```
1.<U(UQ(◊C(◊F N(ΠD C1(◊E($GEBIET)))◊Y)SIT(ΔB))ZEIT(ΔA))>.
  DAB.:<U(UQ(S(N(ΠD C1($GEBIET))C1($GEOGR))SIT(ΔB))ZEIT(ΔA))>.
  --->
  <U(UQ(◊C(◊F N(ΠD C1(◊E($RE1GION)))◊Y)SIT(ΔB))ZEIT(ΔA))>.

2.<U(UQ(◊C(◊F N(ΠD C1(◊E($GEBIET)))◊Y)SIT(ΔB))ZEIT(ΔA))>.
  DAB.:<U(UQ(S(N(ΠD C1($GEBIET))C1($WISSENSCH))SIT(ΔB))ZEIT(ΔA))>.
  --->
  <U(UQ(◊C(◊F N(ΠD C1(◊E($DOMAINE)))◊Y)SIT(ΔB))ZEIT(ΔA))>.
```

As the linear notation is rather difficult to read, mainly owing to the non-terminal variables "◊C" and "◊E", we give the left-hand "structural description", which is the same for both rules and is, moreover, identical to their right-hand sides (with the exception of the semanteme), in tree notation.

```
U-------------------------------------
I                                    I
I                                    I
UQ----------------------------       ZEIT
I                            I       I
I                            I       I
◊C----------------------     SIT     ΔA
I      I               I     I
I      I               I     I
◊F     N-------        ◊Y    ΔB
       I       I
       I       I
       ΠD      C1
               I
               I
               ◊E
               I
               I
               $GEBIET
```

Fig.9 Structural description of the transfer rules

This structural description applies to any object tree with "$GEBIET" occurring somewhere under a node labelled "N", e.g. to the deep structure of the sample sentence (Fig.8), since the substructures assigned to the "◊" variables may be empty. Note that the different occurrences of the variable "¤D" (in the three components of both rules) stand for identical quantifiers. The semantemes appearing in the structural restrictions, "$GEOGR" and "$WISSENSCH", correspond to English "geographic" and "scientific" respectively.

The two occurrences of "$GEBIET" in each rule have to be referentially identical. In the present version of the rules this is only imperfectly guaranteed by the identity of situation and time of utterance (expressed by identical variables)[11].

5.2 Data Base and Deduction Rules

We assume the sample data base to contain the following meaning postulates, i.e. deep structures valid for the "universal situation" and the "universal time" (which is expressed by "$ALLS" and "$ALLZ"), given in linear notation:

```
1.<U(UQ(S(N(Q($ALL)C1($MISCHPULT))C1($ELEKTRON))
  SIT($ALLS))ZEIT($ALLZ))>.
```
("Every sound mixer is something electronic.")

```
2.<U(UQ(S(N(Q($ALL)C1($ELEKTRON))C1($TECHN))
  SIT($ALLS))ZEIT($ALLZ))>.
```
("Everything electronic is something technical.")

```
3.<U(UQ(S(N(Q($ALL)C1($TECHN))C1($WISSENSCH))
  SIT($ALLS))ZEIT($ALLZ))>.
```
("Everything technical is something scientific.")

```
4.<U(UQ(S(N(Q($ALL)C1($ALPIN))C1($GEOGR))
  SIT($ALLS))ZEIT($ALLZ))>.
```
("Everything Alpine is something geographic.")

Only the last of these meaning postulates is relevant for our example.

The present version of DEDUKT uses two kinds of deduction rules:

- "clause rules" that are applied to the deep structures assigned to translated sentences and that split them up into suitable pieces of information ("clauses");
- "inference rules" that are applied to data-base structures and to intermediate results of deduction.

The example was carried out with the following set of clause rules:

```
1.<S(◊G(N(Q(ΔA)C1(MC1(KC1($UND)C1(¤B))C1(¤C)))ΔD))>.
  --->
  <S(N(Q(ΔA)C1(¤C))C1(¤B))>.
```
(example: "The green apple is sweet." ---> "The apple is green.")

```
2.<S(N(Q(ΔA)C1(¤B))C1(MC1(KC1($UND)C1(¤C))C1(¤D)))>.
  --->
  <S(N(Q(ΔA)C1(¤B))C1(¤C))>.
```

(example: "The apple is green and sweet." ---> "The apple is green.")

```
3.<S(N(Q(ΔA)C1(¤B))C1(MC1(KC1($UND)C1(¤C))C1(¤D)))>.
  --->
  <S(N(Q(ΔA)C1(¤B))C1(¤D))>.
```

(example: "The apple is green and sweet." ---> "The apple is sweet.")

For the first clause rule to be valid it is, of course, necessary that the specifications of situation and time of utterance be the same for the antecedent and the consequent. This is, in fact, guaranteed: since the corresponding parts of the object tree do not appear in the rule structures, they remain unchanged in the course of the application.

The inference rules of the sample transfer read:

```
1.<U(UQ(S(N(Q($ALL)C1(¤A))C1(¤B))SIT(ΔF))ZEIT(ΔE))>
  <U(UQ(S(N(Q(ΔD)C1(¤C))C1(¤A))SIT(ΔF))ZEIT(ΔE))>
  <ZERO>.
  --->
  <U(UQ(S(N(Q($ALL)C1(¤A))C1(¤B))SIT(ΔF))ZEIT(ΔE))>
  <U(UQ(S(N(Q(ΔD)C1(¤C))C1(¤A))SIT(ΔF))ZEIT(ΔE))>
  <U(UQ(S(N(Q(ΔD)C1(¤C))C1(¤B))SIT(ΔF))ZEIT(ΔE))>.
```

(informally: "If every A is a B at time E in situation F and if some C is an A at time E in situation F, then some C is a B at time E in situation F", where "some" may be replaced by other quantifiers except "no".)

```
2.<U(UQ(S(¤Y)SIT($ALLS))ZEIT($ALLZ))>
  <U(UQ(S(¤Z)SIT(ΔA))ZEIT(ΔB))><ZERO>.
  BED.:<&(¬(CONTA(ΔA $ALLS))¬(CONTA(ΔB $ALLZ)))>.
  --->
  <U(UQ(S(¤Y)SIT($ALLS))ZEIT($ALLZ))>
  <U(UQ(S(¤Z)SIT(ΔA))ZEIT(ΔB))>
  <U(UQ(S(¤Y)SIT(ΔA))ZEIT(ΔB))>.
```

(informally: "If a sentence Y is true at all times and in all situations, Y is also true for time B and situation A which are the utterance parameters of sentence Z.")

Both inference rules are three-place transformational rules with two structures (representing the antecedents) occurring identically on both sides and a third structure (which represents the result proper of the inference) replacing the empty structure "<ZERO>". The double occurrence of the first two structures is due to the fact that they must not be deleted in the course of rule application; for they are to remain in the data base. "<ZERO>" is needed on the left-hand sides, because the transformational formalism of SALAT requires the rules to have the same number of structures on both sides.

The structural restriction of the second inference rule is to be read as follows: the substructure assigned to "ΔA" (in applying the rule to an object tree) does not contain "$ALLS" and the substructure assigned to "ΔB" does not contain "$ALLZ". As in the present version of the deep syntax the categories "SIT" and "ZEIT" cannot dominate more than one node each, this is equivalent to: the node assigned to "ΔA" is not labelled "$ALLS" and the node assigned to "ΔB" is not labelled "$ALLZ". This condition is necessary, for if sentence Z were universally valid, too, no new structure would be generated (the third structure being identical with the first).

The second antecedent structure of the second inference rule, which might at a first glance seem superfluous, is necessary for two reasons: firstly, without it no value could be assigned to the variables occurring in the structural restriction, and secondly, it would be useless to relate the universally valid structure to a situation and a time that do not appear in any other structure, for then no further deduction would be rendered possible (compare the sample deductions in 5.3).

5.3 Steps of Deduction

Regarding the data of the example (sample sentence, transfer rules, data base, and deduction rules), it is intuitively obvious, which deductions ought to be carried out for the choice between the two potential transfer rules: since the referent of "$GEBIET" has the property "$ALPIN" and since "$ALPIN" implies "$GEOGR" according to the last data-base expression, the "$GEBIET" is "$GEOGR", and hence it follows that "$RE1GION" is the TL semanteme to be substituted for "$GEBIET".

Three steps of deduction are carried out by DEDUKT to attain the desired result, two of them being preparatory, as it were. First, the information contained in the object tree (Fig.8) has to be split up into clauses by the clause rules. Only the first of them is applicable. The result of its application (Fig.10) is added to the data base.

```
U----------------------------
|                           |
|                           |
UQ--------------------      ZEIT
|                    |      |
|                    |      |
S--------------      SIT    $Z1
|             |      |
|             |      |
N-------      C1     $S1
|      |      |
|      |      |
Q      C1     $ALPIN
|      |
|      |
$DEF   $GEBIET
```

Fig.10 Result of the first step of deduction

In the next step, DEDUKT applies the second inference rule to the last data-base structure and to the result of the first step. The result proper of the application (i.e. the third output structure, given in Fig.11) states that the universally valid meaning postulate is true at time "$Z1" and in situation "$S1" as well. As the rule operates on data-base structures this result becomes a part of the data base, too.

```
U----------------------------
|                           |
|                           |
UQ--------------------      ZEIT
|                    |      |
|                    |      |
S--------------      SIT    $Z1
|             |      |
|             |      |
N-------      C1     $S1
|      |      |
|      |      |
Q      C1     $GEOGR
|      |
|      |
$ALL   $ALPIN
```

Fig.11 Result of the second step of deduction

The second inference rule is not applied to any other data-base structure, according to the principle (which is implemented in DEDUKT) that data-base expressions are regarded only if they have semantemes in common with the structure to be proved.

After these two preparatory steps the decisive deduction can be carried out: the first inference rule is applied to the results of the first two steps, yielding the new deep structure given in Fig.12.

```
U----------------------------
|                           |
|                           |
UQ--------------------      ZEIT
|                    |      |
|                    |      |
S--------------      SIT    $Z1
|             |      |
|             |      |
N-------      C1     $S1
|      |      |
|      |      |
Q      C1     $GEOGR
|      |
|      |
$DEF   $GEBIET
```

Fig.12 Result of the third step of deduction

This is exactly the argument of "DAB" of the first transfer rule in tree notation, where the variables are, of course, replaced by their actual values stemming from the object tree (Fig.8). So the first transfer rule is applicable, and "$GEBIET" is replaced by "$RE1GION". Since the structural restrictions of the transfer rules for one SL semanteme are assumed to constitute mutually exclusive conditions of application, the first rule found to be applicable is the only one that will apply at all. So, in the case of our example, the second transfer rule need not be tested for applicability, in particular, its "DAB" argument need not be refuted by DEDUKT.

The preceding example has been tested with the present first transfer version of SALAT (cf. chapter 3.6.1.2 of (17)). Needless to repeat that all its components are still in a more or less provisional form. Nevertheless, they will hopefully have contributed to the clarification of the principles respected and methods provided in SALAT for the treatment of some semantic problems in machine translation.

Notes

1 SALAT is being developed by the Project A2 "Automatische Übersetzung" ("Machine Translation") of the Sonderforschungsbereich 99 "Linguistik" at the Institut für Angewandte Sprachwissenschaft (Institute of Applied Linguistics) of the University of Heidelberg. The system is implemented on the IBM 370/168 of the computing center at the University of Heidelberg and is being tested on simple sample sentences. The project is headed by Prof. Dr. Klaus Brockhaus. It is sponsored by the Deutsche Forschungsgemeinschaft, as well as the whole Sonderforschungsbereich, most projects of which are located at the University of Konstanz.

2 For technical reasons it has been impossible to reproduce the original computer output of the examples, but they have been imitated as far as possible by typewriter. Some abbreviations have been introduced (in particular, the canonical node numbers of nearly every structure have been left out) and some special signs had to be replaced by others.

3 This is, of course, only one reading of the German sentence. By the way, an earlier version of the deep structure in Fig.1 occurred in a sample translation with SALAT which is documented and described in some detail in (14). That version of the interlingua did not explicitly represent time and situation of utterance in the deep structures assigned to translated sentences. Properly speaking, it is not sentences, but utterances (roughly: sentences in specific situations of utterance) that are translated. This is reflected in SALAT in the following way: the utterance parameters (presently names of times and situations, more parameters might be needed) of the deep structures may be viewed as pointers to data-base expressions with the same names occurring in them. These expressions contain supplementary information about the utterance

specifications. The foregoing should be kept in mind if we speak of "translated sentences" in this paper.

4 Such a device is very convenient, as, for the purpose of control, the structures are printed out in tree form, and e.g. in the Russian context-free syntax of (10) there are maximally 16 subcategories belonging to one main category.

5 This is, of course, no complete description of relative modification in English. E.g. relative clauses containing neither relative pronouns nor "that" can modify any noun, so they would be specified as "ART(1,2)", which means that they can agree with "ART(1)" nouns as well as with "ART(2)" nouns.

6 This condition is motivated in the following way: it describes the typical position for a verb representation, which is necessary, because there are other positions typical for noun or adjective representations and in these cases the semanteme ought to be substituted by a noun lexeme ("barker") or by an adjective lexeme ("barking"). The decidable predicate of the sample rule holds true e.g. for the deep structure in Fig.1.

7 It is shown in (3) that the proofs usually adduced that natural languages are not context-free do not hold, for the property is proved only for subsets of the grammatical sentences (e.g. for "respectively" constructions), which, by itself, does not imply that the whole set of grammatical sentences has the property of being non-context-free.

8 For instance, the semantemes representing 'logical words' (some conjunctions, negations, quantifiers) belong to the set of 'international' semantemes. If the corresponding lexemes have non-logical readings, too, these have to be represented by other semantemes, which will often be specific to one language only.

9 This principle is motivated by the hypothesis that, in case of conflict, special information is to be preferred to more general information. In particular, information from the context seems to be able to override general knowledge or meaning postulates. The three principles mentioned here are respected in the transfer example (cf. 5.3).

10 "klein" ("small") is represented here as if it were an intersective adjective, but it is in fact a subsective one. This requires a more complicated representation, because a standard of comparison is needed (e.g. it is not the same to be "small" for an elephant or to be "small" for a mouse). The appropriate structure for subsective adjectives is given in (19), pp.6ff.

11 Indeed, this solution is not as bad as it might seem, since each sentence of a translated text gets a distinct specification of utterance time and situation (represented by distinct time and situation names, which may be related to each other

with the aid of suitable data-base expressions). However, there might be cases where e.g. a definite description occurring twice in one sentence had different referents for each occurrence. Thus a satisfactory solution of the problem would require that at least different occurrences of noun phrases be marked by differently specified utterance parameters.

References

1 Bar-Hillel, Yehoshua (1962): "Machine Translation: The End of an Illusion." Information Processing 62. Proceedings of the IFIP Congress 1962, ed. by Cicely Popplewell. Amsterdam: North-Holland Publ. Co. 1963, pp.331-332

2 Bar-Hillel, Yehoshua (1971): "Some Reflections on the Present Outlook for High-Quality Machine Translation." Feasibility Study on Fully Automatic High-Quality Translation. RADC-TR-71-295, Vol.I. December, 1971. Rome, N. Y.: Air Development Center, pp.72-76

3 Brandt Corstius, Hugo (1976): "The Proof that Natural Languag is not Context-Free." Paper read at the International Colloquium of the Sonderforschungsbereich 100 "Elektronische Sprachforschung", 23.9.-25.9.1976 at Saarbrücken. Abstract in the Preprints. Saarbrücken: Universität des Saarlandes, pp.22-23

4 Brockhaus, Klaus (1975): "Ein Syntaxformalismus für Flexion und Wortbildung." Linguistische Berichte 40, pp.45-62.

5 Brockhaus, Klaus (1976): "Benutzungsanleitung für das Programmsystem COSY. Version 2." Working paper B11 of SFB 99/A2. Brockhaus et al. 1976, Vol.I, pp.94-124

6 Brockhaus, Klaus / Hauenschild, Christa / Höfling, Jürgen / Huckert, Edgar / Maier, Robert (1976): Forschungsbericht 1.11.1973-31.3.1976. Sonderforschungsbereich 99 "Linguistik" /Teilprojekt A2 "Automatische Übersetzung". Universität Heidelberg

7 Cresswell, Max J. (1973): Logics and Languages. London: Methuen

8 Dilger, Werner / Schneider, Martin (1976): Automatisierbare und automatisierte Beweisverfahren für die Logik 1. Stufe. Mit Anhängen von Heribert Ficht und Hubert Schleichert. Konstanz: Sonderforschungsbereich 99

9 Egli, Urs (1975): Ansätze zur Integration der Semantik in die Grammatik. Kronberg (Taunus): Scriptor

10 Hauenschild, Christa (1976): "Kontextfreie Satzsyntax für ein Fragment des Russischen." Working paper C7 of SFB 99/A2. Brockhaus et al. 1976, Vol.II, pp.70-240.

11 Hauenschild, Christa (1978): "SALAT: System for Automatic Language Analysis and Translation." Paper read at the 7th International Conference on Computational Linguistics, Bergen (Norway) 14-18 August, 1978. To appear in the Proceedings

12 Hauenschild, Christa (in prep.): Pronominalisierung am Beispiel des Deutschen. Working paper C11 of SFB 99/A2. Heidelberg

13 Hauenschild, Christa (in prep.): Automatische semantische Analyse des Russischen: Artikelrekonstruktion. Diss. Heidelberg

14 Hauenschild, Christa / Huckert, Edgar / Maier, Robert (1978): "SALAT: Entwurf eines automatischen Übersetzungssystems." Working paper CER1 of SFB 99/A2. Heidelberg. To appear in: Sprache und Datenverarbeitung

15 Huckert, Edgar (1976): Kontextfreie Wort- und Satzsyntax für einen Ausschnitt des Französischen. Working paper E12 of SFB 99/A2. Heidelberg

16 Huckert, Edgar (1978): TRANSFO. Ein System zur Erzeugung transformationeller Ableitungen. 2nd, rev. and enl. version of working paper E13 of SFB 99/A2. Heidelberg

17 Huckert, Edgar (in prep.): Automatische Synthese des Französischen aus einer logischen Basis. Diss. Saarbrücken

18 Koutsoudas, Andreas (ed.) (1976): The Application and Ordering of Grammatical Rules. The Hague-Paris: Mouton

19 Maier, Robert (1976): "Die ε-λ-kontextfreie Basis des Übersetzungssystems SALAT." Working paper R3/4 of SFB 99/A2. Brockhaus et al. 1976, Vol.I, pp.45-66

20 Maier, Robert (1978): Zur Syntax und Semantik der Basis von SALAT - Stand Ende 1977. Working paper R13 of SFB 99/A2. Heidelberg

21 Stechow, Arnim von (1974): "ε-λ-kontextfreie Sprachen. Ein Beitrag zu einer natürlichen formalen Semantik." Linguistische Berichte 34, pp.1-33

22 Suppes, Petrick (1973): "Semantics of Context-Free Segments of Natural Language." Hintikka/Moravcsik/Suppes (eds.): Approaches to Natural Language. Proceedings of the 1970 Stanford Workshop on Grammar and Semantics. Dordrecht: Reidel, pp.370-394

23 Thomason, Richmond (ed.) (1974): Formal Philosophy: Selected Papers of Richard Montague. New Haven: Yale Univ. Press

24 Wilks, Yorick (1973): "An Artificial Intelligence Approach to Machine Translation." Schank/Colby (eds.): Computer Models of Thought and Language. San Francisco: Freeman, pp.114-151