Machine Translation: The Languages Network (versus the intermediate language.)

P.C. ROLF
Dpt. of Computational Linguistics
Literary Faculty
Nijmegen University
NL6500 HD Nijmegen
The Netherlands

Abstract

Jonathan Slocum /Slocum, 1985/ divides MT techniques from a linguistic point of view into three two-way perspectives which are not quite disjunct: "direct versus indirect; interlingua versus transfer; and local versus global scope.".
In this paper we present a research paradigm which, in fact, does not exactly match any of these perspectives: The Languages Network.
In this paradigm each pair of languages will be treated as within a transfer application but with the characteristics of indirect translation: analysis of the source language and synthesis of the target language are not totally dependent on each other.
The proces must be split up into a large number of pieces which can be connected into a huge network performing MT from and into several languages.
Implementations of this paradigm are being carried out by the author by means of the translator generator SYGMART (see /Chauché/, /Chauché, 1974/ and /Rolf, 1985/), which permits the linguist to implement whatever he wants in the field of MT in an efficient way on a wide range of computers (from Atari1040STf via SUN's to IBM VM/CMS mainframes).

1 The goal of the intermediate language.

In discussions on translation systems, the question is often asked whether the system is based on direct translation, whether it works according to the transfer method, or whether it uses an intermediate language. This question suggests that translation systems can be defined exactly by dividing them into these three categories.
It is our conviction that solving problems in translation is more complex than is suggested by this question. In this paper we will show precisely where this question is inadequate, by looking at some aspects of the translation process.

1.1 Why an intermediate language?

The idea which leads to the definition of an intermediate language originates from the wish that, in translation from one source language into several target languages, no completely different translation should have to be made for each pair of languages.
Brandt Corstius /Brandt Corstius, 1978/ expresses this idea as follows (the citation has been translated from Dutch into English): "Instead of making 90 programs in order to translate ten languages into one another (from each language into each of the nine others), it would be sufficient to have 20 translation programs (from each language into Machinish, and from Machinish into each of the languages). It is even conceivable that eighteen programs would be sufficient, if one of the ten languages is given the role of this intermediate language."
This standpoint implies an "efficient" method in terms of the amount of work, without indicating whether this method solves also any principal problems with respect to machine translation.
On the contrary, Brandt Corstius remains sceptical about this.

1.2 What shall be defined?

Ideally the intermediate language will have to be an unambiguous representation of the meaning of (each of) the source language(s). This implies that a language should be found in which it is possible to represent all possible meanings in an unambiguous way.
If Brandt Corstius is followed in this, then in the translation from and into natural language this would have to be one of the natural languages. But one of the main characteristics of natural language is precisely that it is efficient, which implies that with few words and constructions a lot can be expressed in very many different circumstances.
The way in which each individual natural language is efficient differs from language to language: ambiguities and vagueness in a source language cannot usually be projected in a one-to-one correspondence onto a target language.
So it would seem to be not entirely plausible to select an intermediate language from the languages to be translated, seeing that the demand of unambiguity is too heavy precisely for natural language. An intermediate language should not only be unambiguous, but it should also be able to represent all possible meanings. We are convinced that for every sentence of a natural language an infinite set of meanings is possible, since meaning depends on the universe of discourse and the set of possible universes of discourse is infinitely large.
All in all there is enough cause for a fundamental approach to the problem: what is to be achieved in defining an intermediate language in the machine translation of a set of natural languages from and into each of the members of that set?
The need for an intermediate language originates, on the one hand, from the idea that the analysis of a source language will be largely the same, irrespective of the target language into wich it is to be translated, and on the other hand, from the need to analyse the source language in such a way that all ambiguities have been solved, and that therefore the generation of the target language can take place without any further problems.
With respect to the former, we, too, believe that the idea that the analysis of a source language is partly the same, irrespective of the selected target language, is entirely correct.
When we plot the translation process on a line from source language to target language, this will be the part which is close to the source language: to put it in rather more linguistic terms: the morphological analysis and that part of the other syntactic analysis that can be summed up in the term surface grammar, so in any case the NP and VP detection, for instance. We shall return to this below.
The second need, viz. completely disambiguating the source language, would seem to be too heavy a demand, as was formulated before, with reference to 'all possible universes of discourse'. The two needs that have been mentioned cannot be fulfilled, but it can be maintained that there is no need for the entire analysis of the source language and the entire generation of the target language to be done over and over again for every pair of languages. It has to be determined what the two paths, analysis and generation, will look like. Parts of these two paths will remain the same, for the source language irrespective of the target language, and for the target language irrespective of the source language.

2 The analysis of the source language.

Since there is no reason to adopt an intermediate language as has been argued, the problem facing us is the analysis of the source language, as well as the generation of the target language and the process between analysis and generation, which will be discussed in the following two sections.

The point of departure for both these sections, and, in fact, for this paper is the hypothesis that the meaning of the source language depends on the objective one has in mind. In the case of machine translation the meaning, expressed in the translation, depends on the target language defined.
In this line of arguing translating is therefore always a matter of a specific relationship between two languages. When we plot the process of translation on a line, however, we can distinguish three phases, which can be referred to as analysis, translation and generation. The present and the next following two sections have been divided on the basis of this principle.

2.1 What is analysis?

The analysis of a source language can be defined, in a very abstract way, as the addition of information to the input. It may seem trivial, but this starting point implies that no information must be lost in the analysing stage. Information may only be added. This also implies that the input order must not be changed. So, in our view, a dependency grammar is not suitable for the analysis because of the loss of the input order.
Changes in the input order can only be brought about on the basis of requirements posed by the target language.
This brings us to a second aspect of the analysing stage: in this stage only source language inherent data are worked with, to be subdivided into static (lexical) and dynamic (grammatical) data.
The fact that in the analysing stage solely data inherent to the source language are used, does not mean that the target language has nothing to do with the nature of the analysis. That would clash with our starting point, viz. that the meaning of the source language depends on the target language.
The influence of the (set of) target language(s) extends over the way in which the analysis is carried out, in other words, what type of information has to be added to the text of the source language.
Let us take, by way of example, the translation Dutch-English. We will assume (for convenience's sake) that in Dutch the word order in subclauses is S(ubject)-O(bject)-V(erb), while in English the standard word order in subclauses is S(ubject)-V(erb)-O(bject).
The change from S-O-V into S-V-O does not belong to the analysing stage of Dutch, for it implies loss of information because of the word order change. However, the translation into English requires from the analysing stage of Dutch that, among other things, the categories S, O and V are assigned. The assignment of S and O implies that NP's have to be found, etc.
The analysing stage comprises all stages which belong to morphology, surface grammar and possibly a large number of matters that belong to the field of semantic interpretation (see /Bakel, 1984/). This last category, semantic interpretation, is close to the translation stage and will possibly be different for groups of target languages. In the section headed 'Prospect' we will indicate schematically how this semantic interpretation has to be situated in the whole of the translation process.

2.2 Algorithmic consequences.

Starting from the assumption that the addition of information (committing abstractions) is brought about, among other things, by the application of some sort of dependency structure, what is needed is a form of graphic representation.
For many languages, and certainly also for Dutch, the traditional tree structure clashes with our demand formulated earlier, that information should be retained: the original word order must be maintained during the analysing stage. (In Dutch a postmodifier in an NP is often extraposed, e.g. "Ik heb **de man** gezien **met de bril**." Translated word by word: "I have **the man** seen **with the spectacles**.")
Furthermore linguists should have the possibility of expressing linguistic notions in a way which is adequate to them. For this purpose a distinction has been made, in the SYGMART system, between the morphological analysis, which operates on words (the subsystem OPALE) and a tree transformational part (the subsystem TELESI), which operates on multi-dimensional trees over text(s) (the notion sentence does not exist in SYGMART).
The surface grammar and the semantic interpretation cannot therefore be algorithmically distinguished.
This multi-dimensionality enables the linguist to establish relationships between sentence constituents which are far apart, without having to extract them out of their original order. This multi-dimensionality has to be looked upon as the definition of graphs more complex than trees over the input. For a more detailed discussion of the multi-dimensionality the reader is referred to /Chauché, 1984/.
Our arguments for not using the traditional grammatical types are given in /Rolf, 1986/. Part of the analysis of Dutch is shown in Appendix A.

3 The translation.

The translation stage is the stage between the source language inherent analysis and the target language inherent generation. This stage can be roughly compared to a transfer component, as suggested in the beginning of section 1.
Two features are characteristic for the translation, viz. word order change and the addition of target language features.
Word order change(s) (better: component or category movement) is (are) not per definition carried out separately for all possible target languages. If in the example of the word order change in subclauses, presented in the previous section, the rule SOV → SVO has to be applied to a subset of the target languages, this can be done for the entire subset prior to the introduction of target language specific features.
This introduction of target language specific features is done by the lexical translation, or the translation of the words. We will assume here that the analysing stage has provided all the necessary information to get the correct translation for every word.
Because of the information added in the analysing stage the correct translation of a word implies the translation of a complex data structure into another complex data structure, in which the written base form in both cases is but one value of that data structure.
On the basis of information that comes in after the lexical translation, further word order changes will generally have to take place, as well as the generation of grammatical structures.
A simple example in this connection is the following: the Dutch verb **blijven** is translated into English **keep**, but in Dutch **blijven** is completed by an infinitive (**blijven wachten**), whereas in English a gerund is expected (**keep waiting**).
If on the basis of the new lexical information further grammatical rules have to be applied, such as moving the verb in the gerund construction (Dutch "ik **blijf** op hem **wachten**" into English "I **keep waiting** for him"), these rules also belong to the translation stage, not to the generation stage, unless the rules apply to all possible source languages with respect to English.
As in the previous section, here, too, the demand made of the algorithmic procedures and the possibility of building and manipulating complex datastructures is heavier than in traditional grammatical types. Within SYGMART the subsystem TELESI is used for the translation stage, which does not imply that for each pair of languages a separate TELESI implementation has to be made after all: SYGMART provides for the application of different TELESI grammars one after another.

4 The generation of the target language.

From the previous sections it has already become apparent that the generation of the target language does not come into play, until only target language inherent matters are at issue, matters which hold irrespective of the source language that is used.
They are in any case all matters of a morphological nature which in the entire translation process are the last to be dealt with. In the translator generator SYGMART the generating morphology is treated by the subsystem AGATE.

If there are grammatical rules which also have to be applied independently of the source language, they also belong to the generating stage, in our set-up. These rules will not be many, for that would imply that in a target language certain constructions should occur for which in no (source) language an analogous construction was to be found.
In our set-up the technical three-way division of SYGMART (OPALE, string into tree, TELESI tree into tree via network, AGATE, tree into string) cannot be measured in a one-to-one correspondence onto the three-way division of the translation process, viz. analyses, translation and generation. The morphological analysis always takes place in OPALE and is a, rather small, part of the entire analysis. The greatest part of the analysis, consequently, takes place in TELESI. Everything belonging to the translation happens in TELESI. As far as the generation is concerned, a small part is possibly carried out in TELESI, but the morphological generation, naturally, takes place in AGATE.

5 Prospect: The Languages Network.

From the foregoing it can be deduced that as far as we are concerned the question formulated in the beginning disregards the complexity of the translation of natural language from and into one another. In general the analysis of the source language is the most important component: once the analysis has been carried out on all possible levels in all possible details, generating the target language is 'relatively' simple: at that stage word meanings have, of course, been disambiguated, semantic interpretation have been assigned, references have been determined, etc., but all this has been done in relation to the meaning, viz. the translation.
Seeing that we analyse on the basis of the requirements of the target language, analysis is only partly an unambiguous notion: not all the abstractions will be equal for all the target languages, nor will they be required for all the target languages.
If for each pair of languages the entire process is plotted on a line from source language to target language, it will be possible to point to a number of points on that line, where analyses will go into different directions (depending on the subset of target languages) and where translations and generations merge. These lines together form the languages network.
On the basis of such a network it will be possible, in the future, to formulate relationships with reference to the affinity of languages mutually. This may sound speculative, but we are convinced that what has been presented cannot be reduced to the definition of a single intermediate language, unless it is done for subsets of natural languages which have been defined precisely. Our objections to this are formulated in /Rolf, 1986/.

6 References.

[Brandt Corstius, 1978]
Brandt corstius, H, **Computer-Taalkunde**, Muiderberg, 1978.

[Bakel, 1984]
Bakel, Jan van, **Automatic Semantic Interpretation**, Foris Publications, Dordrecht, 1984.

[Barr, 1981]
Barr, Avron and Edward a. Feigenbaum, **The Handbook of Artificial Intelligence**, Stanford, 1981, Vol.I.

[Chauché, 1974]
Chauché, Jacques, **Transducteurs & Arborescences, Etudes et réalisations de systèmes appliquées aux grammaires transformationnelles**, Thèse d'Etat, Grenoble, 1974.

[Chauché]
Chauché, J., **Système de manipulations algorithmiques et récursives de texte, SYGMART**, Laboratoire de traitement de l'information, Le Havre, France.

[Chauché, 1982]
Chauché, J., V. Cheboldaeff, R. Lescoeur, M. Jatteau, **Rapport de présentation du project relatif à l'écriture des specifications d'un système de traduction assistée par ordinateur**, SLIGOS, Conseil, assistence et réalisations, produits informatiques, services de gestion, Avril 1982.

[Chauché, 1984]
Chauché, J.,**Un outil multidimensionnel de l'analyse du discours**, in: Proceedings of Coling84, Stanford University, California, pp. 11 - 15, 1984.

[Landsbergen, 1982]
Landsbergen, J., **Machine Translation based on logically isomorphic Montague grammars**, in: Horecky, J. (ed.), COLING 82, pp 175-182, North-Holland Amsterdam, 1982.

[Landsbergen, 1984]
Landsbergen, Jan, **Isomorphic Grammars and their use in the Rosetta Translation System**, Philips Research Laboratories Eindhoven - the Netherlands, (presented at the Tutorial on Machine Translation, April 2 - 6, 1984, Lugano), 1984.

[Rolf, 1983]
Rolf, P.C., **Vertalen van getalsnamen**, Verslagen Computerlinguïstiek No. 3 - 1983, KU Nijmegen.

[Rolf, 1985]
Rolf, P.C., **Beschrijving van het vertaalsysteem SYGMART, versie 4**, intern verslag KU Nijmegen, 1985.

[Rolf, 1986]
Rolf, P.C. and J. Chauché, **Machine Translation and the SYGMART system.**, in: Computers and the Humanities, 1986, Vol.20, no 4,pp 283-288.

[Slocum, 1985]
Slocum, Jonathan, **A Survey of Machine Translation: its History, Current Status, and Future Prospects.** in: Computational Linguistics, Vol 11, Number 1, January-March 1985.

[Witkam, 1983]
Witkam, A.P.M., **Distributed Language Translation**, Feasibility Study of a Multilingual Facility for Videotex Information Networks, BSO, Utrecht, 1983.

A An example of the analysis of Dutch.

The following is an extract of a running implementation of a part of the analysis of Dutch in the SYGMART system. It can be regarded as one of the nodes in the proposed Languages Network.
This node constitutes a network in itself, with four possible entry points, characterised by "&ENTREE:", and several end nodes, characterised by "-- >%STOP.".

```
  &GRAMMAIRE.

/*  Basisgrammatica's  */

&GRAM: NP(E).
  FINDKERN: O(1(*))    /
     1: ((WRDSOORT=SUBST)|(PRON=PERS))&(CATEGORY^=NCKERN)
  =>  X(Y(1)) /
     X: (CATEGORY=NBAR) ;
     Y: (CATEGORY=NCKERN).

  FINAPQP2: O(1(2),*,3) /
     2: (WRDSOORT=ADJ)|(WRDSOORT=TELW);
     3: CATEGORY=NBAR
  =>  X(*0<,1>*,Y(1(2),*3<,>*),*0<3,>*)     /
     X: 0 ;
     Y: 3 ;
     1: 1(SUBCAT(2)).
```

```
RESTADJ:  0(1(2(*)),*,3(4)) /
   2: WRDSOORT=ADJ;
   3: CATEGORY=NBAR;
   4: CATEGORY=AP
=>  X(*0<,1>*,Y(Z(2,*4<,>*),*3<4,>*),*0<3,>*)  /
   X: 0 ;
   Y: 3 ;
   Z: 4.

FINDEQP1: 0(1(2),*,3) /
   2: (WRDSOORT=LIDW)|(WRDSOORT=TELW)   /
   LOCDEQP1(0,2,3)
=>  X(*0<,1>*,Y(1(2),3),*0<3,>*)   /
   X: 0 ;
   Y: (KENCAT(3)) ;
   1: 1(KENCAT(2)).

RESDEQP1:   0(%1(%2),*,3(4))  /
   0: (VORM='SENTENCE*****')|(VORM='BIJZIN*****')|
      (CATEGORY=PVWWCL);
   2: (WRDSOORT^=LIDW)&(WRDSOORT^=TELW)   /
       CATRES(3,4)
=>  X(*0<,1>*,%1(%2),Y(3(4)),*0<3,>*)  /
   X: 0 ;
   Y: (KENCAT(3)).

ONLYADJ:   0(1(2(*)))  /
   0: (VORM='SENTENCE*****')|(VORM='BIJZIN*****')|
      (CATEGORY=PVWWCL);
   1: CATEGORY^=AP;
   2: WRDSOORT=ADJ
=>  X(*0<,1>*,Y(1(2),KERN(EMPT)),*0<1,>*)   /
   X: 0;
   Y: (CATEGORY=NBAR);
   1: 1(SUBCAT(2));
   KERN: (CATEGORY=NCKERN).
-->%NUL.

&GRAM: NPCONS(E).
  PREPCON:  0(1(2),*,3)  /
   2: WRDSOORT=PREP ;
   3: CATEGORY=NP
=>   X(*0<,1>*,Y(2,3),*0<3,>*)   /
   X: 0 ;
   Y: *PREP.
--> %NUL.

&GRAM: WWCLUS(E).
  FINDWWCL:  TOP(1,*,2)  /
   TOP: VORM='SENTENCE*****';
   1: (WERKW^=WERKW->0)&(WERKW^=PV)&(WERKW^=PV|IMPERAT);
   2: (WERKW^=WERKW->0)&(WERKW^=PV)&(WERKW^=PV|IMPERAT)   /
      WWCL(1,2)
=>  NEWTOP(*TOP<,1>*,NEW(1,2),*TOP<2,>*)  /
   NEWTOP: TOP;
   NEW: (CATEGORY=WWCL;VERBUM=VERBUM(1));
   2: (KENWW(1,2)).

  FIBYPVCL:  TOP(1,*,2)  /
   TOP: VORM='SENTENCE*****';
     1: (WERKW@>=PV);
     2: (WERKW^=WERKW->0)  /
         PVWWCL(1,2)
  =>  NEWTOP(*TOP<,1>*,NEW(1,2),*TOP<2,>*)  /
     NEWTOP: TOP;
     1: 1(WERKW=PV);
     NEW: *PVWWCL;
     2: (KENWW(1,2)).

   FBYPVCL2:  TOP(2,*,1)  /
     TOP: VORM='SENTENCE*****';
     1: (WERKW@>=PV);
     2: (WERKW^=WERKW->0)&(CATEGORY^=WWCL)  /
          PVWWCL(1,2)
  =>  NEWTOP(*TOP<,2>*,NEW(2,1),*TOP<1,>*)  /
     NEWTOP: TOP;
     1: 1(WERKW=PV);
     NEW: *PVWWCL;
     2: (KENWW(1,2)).

  FINDPVCL:  TOP(1,2)  /
     TOP: VORM='SENTENCE*****';
     1: (WERKW@>=PV);
     2: (WERKW^=WERKW->0)  /
          PVWWCL(1,2)
  =>  NEWTOP(*TOP<,1>*,NEW(1,*TOP<1,2>*,2),*TOP<2,>*)  /
     NEWTOP: TOP;
     1: 1(WERKW=PV);
     NEW: *PVWWCL;
     2: (KENWW(1,2)).

  FINDPV: TOP(1)  /
     TOP: VORM='SENTENCE*****';
     1: (WERKW@>=PV)
  =>  NEWTOP(*TOP<,1>*,NEW(NEW1(1)),*TOP<1,>*)  /
     NEWTOP: TOP;
     NEW: (CATEGORY=PV);
     NEW1:  (WERKW=PV;
         <(VERBSORT(1)^@>=COPULA): VERBSORT=ZELFST  #
                                   VERBSORT=COPULA  >).

-->%NUL.

&GRAM: BYZIN(E).
  BYZBEG:  TOP(*)   /
     TOP: (PRON=RELPRON)|(VOEGWRD=ONDER)
  =>  NEWTOP(NEW2(TOP))   /
     NEWTOP: (VORM='BIJZIN*****';PRON=PRON(TOP)).

  BYZREST:  0(1(2),*,3)   /
     2: VORM='BIJZIN*****';
     3: (CATEGORY^=WWCL)&(CATEGORY^=PVWWCL)&(CATEGORY^=PV)
  =>  NEWTOP(*0<,1>*,1(NEW2(*2<,>*,3)),*0<3,>*)  /
     NEWTOP: 0;
     NEW2: 2.

  BYZEND:  0(1(2),*,3)   /
     2: VORM='BIJZIN*****';
     3: (CATEGORY=PV)|(CATEGORY=PVWWCL)|(CATEGORY=WWCL)
  =>  NEWTOP(*0<,1>*,1(NEW2(*2<,>*,3)),*0<3,>*)  /
     NEWTOP: 0;
     NEW2: 2.

  RELBYNP:  TOP(1,*,3(4))  /
     1: CATEGORY=NP;
     4: (VORM='BIJZIN*****')&(PRON=RELPRON)
  =>  NEWTOP(*TOP<,1>*,1(3(4)),*TOP<3,>*)  /
     NEWTOP: TOP.
--> %NUL.

&ENTREE: TELWRD(I).
  DELTIGEN:   0(1(*),*,2(*))    /
     2: ((VORM='tig')|(VORM='en'))
  =>  0(1)    /
     1: 1(<VORM(2)='tig':SOORT=SOORT(2)>).

  KEEROM:  0(1(*),*,2(*))  /
     1: SOORT=EENHEID ;
     2: SOORT=TIENTAL
  =>  0(2,1).

  DEL_ENHO: 0(1(*),*,2(*),*,3(*)) /
     1: (SOORT=DUIZTAL);
     2: (SOORT=EENHEID);
     3: (SOORT=HONDTAL)
  =>  0(1,3) /
     3: 3(REPR=REPR(2)).

  VULOP: 0(1(*),*,2(*))  /
     0: (WRDSOORT=TELW)   /
        AANW(1,2)
  =>  0(1,3,2)  /
     3: (KENTEL(1)).

  VULAAN: 0(1(*),*)  /
     1: (SOORT=DUIZTAL)|(SOORT=HONDTAL)|(SOORT=TIENTAL)
  =>  0(1,2)  /
     2: (KENTEL(1)).
```

```
DEL_DU(*(*);0(2)): 0(1(*),*,2(*)) /
    1:(SOORT=EENHEID)|(SOORT=COMBI);
    2: (SOORT=DUIZTAL)|(SOORT=HONDTAL)
 => 0(2) /
    2: 2(REPR=REPR(1)).
--> %STOP.

&ENTREE: ENGTLWD(I).
 Transtel$HLT(Transtel): blad(*)  /
   blad: (TRANS=TRANS->0)
 => blad   /
   blad: blad(TRANS=TRANS(DICT(*))).

 CHNGORD:  0(een(*),*,en(*),*,tien(*),*,tig(*))  /
    een: SOORT=EENHEID ;
    en:  VORM='en';
    tien: (SOORT=TIENTAL)|(SOORT=EENHEID);
    tig:  VORM='tig'
 =>  0(tien,tig,een).

--> %STOP.

&GRAM: RESTNP(I).
 RESTADJ.
 ONLYADJ.
 FINAPQP2(*(*);Y(2)).
 FINDEQP1(*(*);Y(2)).
 RESDEQP1(*(*);Y(4)).
--> %STOP.

&GRAM:  HOUDOP(E).
--> %STOP.



/* Netwerk van grammatica's */

&ENTREE: SEPARATE(I).

 ONBEKEND(*(*);0(1))$TRF(HOUDOP): 0(1) /
     1: DICT@>=ONBEKEND
 =>  0(1).

 LEEST$HLT(ONBEKEND):     1(2(4(*),*,3(*)))  /
     3: LEEST^=LEEST->0
 =>  NEW1(*1<,2>*,4),NEW2(3,*1<2,>*)  /
     NEW1: 1.

 LEEST2$HLT(ONBEKEND):     1(2(4(*),*,3(*)))  /
     4: LEEST^=LEEST->0
 =>  NEW1(*1<,2>*,4),NEW2(3,*1<2,>*)  /
     NEW1: 1.

 SPLITS(*(*);X(P2,2))$HLT(LEEST,LEEST2,ONBEKEND):
     0(1(P1),2(P2))   /
     P1: LEEST=EOSENT;
     P2: LEEST=EOSENT
 =>  Y(*0<,1>*,1(P1),X(*0<1,2>*,2(P2)),*0<2,>*)  /
     X: (VORM='SENTENCE*****') ;
     Y: 0.

 SPLITOP$HLT: 0(1(P))  /
     P: LEEST=EOSENT
 =>  Y(X(*0<,1>*,1(P)),*0<1,>*)  /
     Y: 0;
     X: (VORM='SENTENCE*****').
--> PREPROC.

&ENTREE: PREPROC(U,PREPROC,PREPROC).
 WW2(*(*);0(2)):      0(1(3(2(*))))  /
     0: VORM='SENTENCE*****';
     1: WRDSOORT=WRDSOORT->0;
     2: WRDSOORT=VERB
 =>  0(1(3(2)))  /
     1: 1(VERBUM=VERBUM(2)).
```

```
 WW1(*(*);1(2)):      1(2(*))  /
     1: WRDSOORT=WRDSOORT->0;
     2: WRDSOORT=VERB
 =>  1(2)  /
     1: 1(VERBUM=VERBUM(2)).

 TEINF(*(*);TOP(4,2)):    0(1(2),*,3(4))  /
     2: VORM='te';
     4: WERKW@>=INF
 =>  TOP(*0<,1>*,NEWTOP(X(2,4)),*0<3,>*)  /
     TOP: 0;
     NEWTOP: (WRDSOORT=WRDSOORT(4);WERKW=TEINF;
              VERBSORT=VERBSORT(4));
     2: 2(WRDSOORT=WRDSOORT->0);
     4: 4(WERKW=INF);
     X: (WERKW=TEINF;VERBSORT=VERBSORT(4)).
--> TELW: 0(*) / 0: (WRDSOORT=TELW)&(SOORT^=CYFER)
                  &(SOORT^=SOORT->0) .
--> FINDNPCL.

&GRAM: TELW(E).
 TELWORD(@TELWRD;0):    0(1(*))   /
     1: (WRDSOORT=TELW)&(SOORT^=CYFER)&(SOORT^=SOORT->0)
 =>   0(1).
--> FINDNPCL.
--> %NUL.

&GRAM: FINDNPCL(I).
 FINDKERN(*(*);X(1))$HLT(FINDKERN).
 FINDWWCL(*(*);NEW(2,1)).
 FIBYPVCL(*(*);NEW(2,1)).
 FBYPVCL2(*(*);NEW(2,1)).
 FINDPVCL(*(*);NEW(2,1)).
 RESTADJ.
 FINAPQP2(*(*);Y(2)).
 FINDEQP1(*(*);Y(2)).
 RESDEQP1(*(*);Y(4)).
 FINDPV(*(*);NEWTOP(NEW)).
 ONLYADJ(*(*);X(EMPT,KERN,2)).
--> FIBYZ: 0 / 0: (PRON=RELPRON)|(VOEGWRD=ONDER).
--> RELBYZIN: 0 / 0: (WRDSOORT=PREP).
--> TRANSLAT.

&GRAM: FIBYZ(I).
 BYZBEG$HLT(BYZBEG).
 BYZEND(*(*);NEWTOP(3,NEW2)).
 BYZREST(*(*);NEWTOP(3)).
--> RELBYZIN: 0 / 0: (PRON=RELPRON)|(WRDSOORT=PREP).
--> TRANSLAT.

&GRAM: RELBYZIN(I).
 RELBYNP.
 PREPCON(*(*);X(3)).
--> TRANSLAT.

&GRAM: TRANSLAT(U,TRANSLAT,TRANSLAT):
     <STAM^=STAM->0 :TRANS=TRANS(DICT(*))>.
--> %STOP.

 &FIN.
```