DOCUMENT RESUME

ED 105 716 FL 006 235

AUTHOR	Lehmann, Winifred P.; Stachowitz, Rolf
TITLE	Development of German-English Machine Translation System.
INSTITUTION	Texas Univ., Austin.
SPONS AGENCY	Rome Air Development Center, Griffiss AFB, N.Y.
REPORT NO	RADC-TR-73-309
PUB DATE	Oct 73
NOTE	64p.; Best Copy Available, some pages may reproduce poorly due to size cf print
EDRS PRICE	MF-$0.76 HC-$3.32 PLUS POSTAGE
DESCRIPTORS	*Computational Linguistics; *Contrastive Linguistics; Deep Structure; Descriptive Linguistics; English; Generative Grammar; *German; Lexicography; *Machine Translation; *Programing; Structural Analysis; Surface Structure; Syntax; Transformations (Language)

ABSTRACT

This report documents efforts over a five-month period toward completion of a pilot system for machine translation of German scientific and technical literature into English. The report is divided into three areas: grammar formalism, programming, and linguistics. Work on grammar formalism concentrated mainly on increasing the power of the subscript grammar to prevent intermediate "forced" readings. Work in system construction concentrated on the completion of the grammar maintenance programs and on the core of the systems programs used by all analysis and production algorithms. The linguistic work concentrated on the coverage of the German surface syntax, the "choice rules" for the generation of the corresponding deep structures and their grammatical description. (Author/AM)

RADC-TR-73-309
Final Technical Report
October 1973

DEVELOPMENT OF GERMAN-ENGLISH MACHINE TRANSLATION SYSTEM

The University of Texas at Austin

Rome Air Development Center
Air Force Systems Command
Griffiss Air Force Base, New York

DEVELOPMENT OF GERMAN-ENGLISH MACHINE TRANSLATION SYSTEM

W. P. Lehmann
R. A. Stachowitz

The University of Texas at Austin

FOREWORD

This technical report has been prepared by The University of Texas at Austin, Box 7247, University Station, Austin, Texas 78712 under Contract F30602-73-C-0192, Job Order No. IDHS0414, covering the period 5 February to 4 July 1973. The RADC project engineer was Mr. Zbigniew L. Pankowicz (IRDT).

This report has been reviewed by the Office of Information, RADC and approved for release to the National Technical Information Service (NTIS).

This technical report has been reviewed and is approved.

APPROVED: Zbigniew L. Pankowicz
ZBIGNIEW L. PANKOWICZ
Project Engineer

APPROVED: H Davis
HOWARD DAVIS
TECHNICAL DIRECTOR
INTEL&RECON DIVISION

FOR THE COMMANDER: Carlo P. Crocetti
CARLO P. CROCETTI
Chief, Plans Office

ABSTRACT

Work towards the completion of a pilot system for machine translation of German scientific and technical literature into English is described. This report describes efforts performed in the area of grammar formalism, programming, and linguistics during the period from February 5, 1973 through July 5, 1973; it supplements the work performed under contract F30602-70-C-0118 [1,2,3].

Work on grammar formalism concentrated mainly on increasing the power of the subscript grammar to permit the prevention of intermediate "forced" readings. Work in system construction concentrated on the completion of the grammar maintenance programs and on the core of the systems programs used by all analysis and production algorithms. The linguistic work concentrated on the coverage of the German surface syntax, the "choice rules" for the generation of the corresponding standard structures (deep structures), and their grammatical description.

TABLE OF CONTENTS

I. *GRAMMAR FORMALISM*

1.1	Word Grammar	*I-1*
1.2	Syntactic Grammar	*I-3*
1.3	Syntactic Choice	*I-3*

II. *SYSTEMS*

2.1	Maintenance Programs	*II-4*
2.2	Systems Programs	*II-9*

III. *LINGUISTICS*

3.1	German Syntax	*III-15*
3.2	Choice Rules	*III-26*
3.3	Standard Grammar	*III-32*
3.4	Lexicography	*III-37*

CONCLUSION 51

References 52

INTRODUCTION

Any attempt to carry out fully automatic quality translation of random texts has to take into account the various linguistic problems confronting mechanical analysis of natural language and must provide solutions for them, without any pre- or post-editing.

The typical problems arising in the analysis and translation of German text into English are

a. discontinuous constructions, in particular verb prefix combinations,

b. idiomatic expressions,

c. empty words, which are not translated,

d. deleted terms with or without meaning change,

e. lexical collocations with internal variable slots, dependent on or independent of environment,

f. category changes in translation,

g. sentence pattern changes in translation,

h. translation of the definite article in cases of inalienable property,

i. ambiguity resolution based on information within a sentence.

In addition, the following problems which pertain to sequences of sentences have to be accounted for and solved:

j. anaphoric relations, in particular pronominal-reference,

k. ambiguity resolution based on information in co-text.

We finally expect a quality mechanical translation system to

l. preserve input non-ambiguity.

The following German sentences provide examples of the difficulties which are encountered and need to be solved. In each of the German examples, the word or word combinations and their English correspondences representing the linguistic problem will be underlined; deleted terms are represented by (), internal variable slots by []. Some of the German sentences will first be followed by a literal translation into English and then by the correct translation.

a. *Er fängt mit dem Experiment an.*

He catches with the experiment at.

He begins with the experiment.

b. *Er liess ihnen Bescheid sagen, dass ...*

He let them notice say that ...

He sent word to them that ...

c. *Die Lösung, an der sie ja lange gearbeitet hatten, wurde schliesslich gefunden.*

The solution, which they had yes worked on for a long time, was finally found.

The solution, which they had worked on for a long time, was finally found.

d. *Die Sonne geht im Osten auf und () im Westen unter.*

The sun goes in the east up and () in the west down.

The sun rises in the east and sets in the west.

Fritz ist nach Spanien () und seine Frau () nach Italien gereist.

Fritz is to Spain () and his wife traveled to Italy.

Fritz traveled to Spain and his wife to Italy.

e. *Die Entwicklung nahm [ihren] Anfang.*

The development took [its] inception.

The development began.

Er traf [keine] Anstalten, zu ...

He met [no] institutions to ...

He made no preparations to ...

f. *Sie experimentierten weiter.*

They experimented further.

They continued to experiment.

g. *Dieser Umstand kam ihnen zu Hilfe.*

This fact came to them to assistance.

This fact came to their assistance.

h. *Er kreuzte die Arme.*

He crosses the arms.

He crosses his arms.

i. *Die Leitung zerbrach.*

The line broke. (Not: The management broke.)

Die Leitung kontrollierte das Unternehmen.

The management controlled the venture. (Not: The line controlled the venture.)

j. *Er ist gelungen. (Der Versuch)*

He has succeeded. (The experiment)

k. *Wir kontrollierten die Leitung. Sie war zerbrochen.*

We checked the line. It was broken. (Not: We checked the management. It was broken.)

Wir kontrollierten die Leitung. Sie war korrupt.

We checked the management. It was corrupt. (Not: We checked the line. It was corrupt.)

l. *Der Motor der Maschine ist zerbrochen. Wir werden sie zurückschicken.*

The motor of the engine is broken. We shall sent it back.

The motor of the engine is broken. We shall send the engine back.

The solution of the linguistic problems exemplified by the sentences above requires a particular approach which has been an integral part of linguistic theory for almost two decades: the reduction of surface texts to intermediate, simpler structures, generally called kernel strings or deep structures. Past experience has shown that quality mechanical translation cannot be obtained by trying to map input surface strings directly into output surface strings or by trying to derive the meaning of a text directly from its surface form.

The Linguistics Research System (LRS) has been designed to overcome and solve such linguistic problems; LRS incorporates the components of current linguistic theory, such as surface component, deep component, transformational component, and semantic component.

The work described in this report complements the description of the efforts towards the attainment of Quality Mechanical Translation in the areas of theory, programming and linguistics given in earlier reports [1,2,3].

SECTION I

GRAMMAR FORMALISM

During the five months' contract period a small number of changes pertaining to the rule format were made. Certain of these have already been described in our Final Report [4]; others were added to increase the linguists' ability to reduce the number of intermediate analyses. The changes pertain to the word grammar, the syntactic grammar, and the choice rules.

1.1 Word Grammar

1.1.1 The M-Operator (Marginal-Symbol Operator)

The M-Operator allows the linguist to establish potential sentence boundaries. Thus, rule C 143

```
C 143     V WORD     V BLANK     V CONJ     V BLANK
          M 2,4                  $ CJ(S)    B
                                 B
```

associates a potential sentence boundary with the blank spaces preceding and following a sentence-conjoining conjunction. The expression M 2,4 is to be read as "insert potential sentence boundary into the second and fourth rule term".

Clause rules are sensitive to such boundaries and only apply to text spans bounded by them.

1.1.2 The I-Operator (Insert Operator)

By means of the insert operator, the linguist can add subscripts and values to a constituent. These can be referred to during syntactic analysis and can be used to restrict the number of possible intermediate interpretations or to avoid readings which would eventually not be well-formed.

Example:

```
C 162     V WORD     V DET          V ADJ
          I 2PA      $ GD           $ GD
          I 3FD      $ CA           $ CA
                     $ NU           $ NU
                     ? IN           $ IN
                     . 2.1,3.1      F
                     . 2.2,3.2
                     . 2.3,3.3
                     . 2.4,3.4
                     P
```

The request I 2PA stands for "add the subscript PA ("precedes adjective") to the second term"; I 3FD, for "add the subscript FD ("follows determiner") to the third term".

1.1.3 The P-Operator (Preference Operator)

The P-Operator allows the linguist to select one reading from multiple interpretations of a particular text span. The following two rules illustrate the occurrence of such multiple interpretations:

```
C 129     V AJ        V VB
          + CL(P,C,S  $ FM(PAPL)
            T,EST)    $ FO(A)
          + FOX(A)    $ TO(*R)
          + TOX(HU,A  $ TS
            L,PL,IN,
            NT,AB)
          $ 2.1VB
          = 2

C 6       V A         V VB
          + CL(P,C,S  $ FM(PAPL)
            T,EST)    $ FO(*A)
          $ 2.2       $ TO
          $ 2.3       $ TS
          $ 2.4TM
          $ 2.1VB
```

Rule C 6 interprets adjectival past participles of intransitive verbs as adjectives, as for example in *der gefallene Schnee*. Rule C 129 interprets those of transitive verbs, e.g., *das durchgeführte Experiment*.

Verbs which occur with an optional accusative object, as in *das gelesene Buch*, will consequently receive two interpretations, although only the latter interpretation is correct. After the application of rules C 14 and C 28 only the correct interpretation of the past participle is retained.

```
C 14      V A        V AJ
          + OX       $ FOX
          $.2.6TM    $ TOX
          $*2.5FO    $ FO
          $*2.6TO    $TTO
          Λ 2        - 2.1,2.3
                     - 2.2,2.4

C 28      V WORD     V LB       V ADJ     V RB
          P                     $ OX      B
                                B
```

1.2 Syntactic Grammar

1.2.1 Arguments of Operations

The capabilities of the analysis algorithms were increased to permit operations between subscripts of the same term. As rules C 129 and C 14 above show, it is thus possible, for example, to extract the semantic values associated with the accusative object governed by a verb [FO(A)] and make them the semantic features of the noun to be modified by the adjective [TM].

1.2.2 Negation of Conjunctions and Disjunctions of Values

The capability of expressing "negate" and "ignore" for conjunctions and disjunctions of values was added. Such expressions have the format *:(value-connector-value [connector-value]): (The terms in brackets indicate repetitions of connector value combinations.) If value combinations are to be ignored, the asterisk is replaced by a minus sign.

1.3 Syntactic Choice

It is frequently possible to determine, during syntactic choice, that the span analyzed by a particular rule will also have been analyzed by another rule. Since syntactic choice makes use of semantic information, a decision between the two rules can often be made. If a rule could thus be rejected by means of the nth decision in a choice rule, it had been necessary, so far, to ask in the main rule whether the nth decision was true. If so, the rule was deleted; if not, it was retained.

The formalism was extended to allow the linguist to state the rejection of the main rule directly in the choice rule. It is thus no longer necessary to ask in the main rule whether the nth decision was true. The rejection is effected by the statement C (conditions), T/E, or F/E; the conditions in parenthesis are optional. The "return to the main rule" (cf. Final Report [3]) is now expressed by T/R or F/R.

SECTION II

SYSTEMS

Certain of the programs described below were begun under contract F30602-70-C-0118; the others were written completely during the five months' reporting period. (The programs completed under the previous contract period are described in [1,2,3].)

During this reporting period the programming effort was divided into two areas: grammar maintenance programs and systems programs.

2.1 Maintenance Programs

2.1.1 Rule Modification

The capabilities of SUBGRM [1] were extended to permit the modification of individual parts of a rule.

The rule modification portion of SUBGRM alters a specified rule according to the instructions of the command.

A modify [M] command has the following format:

M RN(.DN), where

RN = rule number
DN = duplication number.

This command is followed by any number of insert [I] or delete [D] instructions. These instructions refer to the exact portion of the rule that is to be altered.

The particular portions of a rule that can be modified are: the rule number, one or more complete terms, or any portion of a particular term—its subscripts (including operators, subscript name, and values), one or more of its operators (not to be confused with subscript operators), and any portion (line) of its choice sets.

The modified rule is then written on a file to be later put on one of the grammar tapes.

2.1.2 Conflation of Dictionary Rules

German and English verb entries are entered by the linguists into the grammar as often as the entry has different meanings and/or selection restrictions. This facilitates the actual encoding and updating of the verb dictionary. During grammar compilation the various entries of one verb are conflated to a single entry to reduce the size of the necessary storage and the number of internal analyses. The conflation is performed by the dictionary rule conflation program.

This program takes on dictionary rules, R_1, R_2,...R_m, each with the same rule number, left-side category symbol, and right-side, where each R_i has a set of left-side subscripts with N columns, and constructs one rule having $T = \sum_{i=1}^{M} N_i$ columns.

A list of subscripts is input to the program and only the subscripts appearing in this list are columnized (each column being separated by an apostrophe). If a columnized subscript is missing in one of the R_i rules, a LA is inserted as the column value. If a subscript is not to be columnized, it is conflated to a single entry and each of the R_i subscripts, if they differ, are separated by commas.

Any trace information contained in individual rules is carried over to the conflated rule.

For example:

```
C 4082     V V          * BEWEG
C 1        + CL(56)
           + PX(FORT)/
           + FS(N)/
           + TS(IN)/
           + FO(A)/
           + TO(R)
           T 1.2
C 4082     V V          * BEWEG
D 2        + CL(56)
           + PX(VORW)/
           + FS(N)/
           + TS(AL)/
           + FO(A)/
           + TO(R)/
           + OA(DIR)
           T 1.2
           T 1.6
```

The subscripts to be columnized are PX, FS, TS, FO, TO, and OA. The resulting conflated rule is:

```
C 4082    V V          * BEWEG
          + CL(56)
          + PX(FORT≠VORW)/
          + FS(N≠N)/
          + TS(IN≠AL)/
          + FO(A≠A)/
          + TO(R≠R)/
          + OA(LA≠DIR)
          T 1.2
          T 1.6
```

2.1.3 CRLNIMG (Create Line Image)

CRLNIMG is the inverse of SUBGRM [1] which converts rules from "print image" to "unpacked format".

CRLNIMG constructs the "print image of a rule" from its "unpacked format". The "print image of a rule" is the form in which the rule is originally coded by the linguists. The "unpacked format" is the internal representation of the rule. All programs which display rules use this routine.

CRLNIMG is flexible in that it allows the calling program to specify the form of the print image in regard to spacing between terms and the width of each term in characters.

2.1.4 GRMDIS2 (Select Symbols with Particular Features)

Purpose:

Read in one or more tapes containing distinct or mixed grammars. If they are mixed, the program separates the rules so that they may be used as distinct grammars. Distinct grammars may also be displayed together.

A display is generated for each request. A request consists of a grammar or grammars with or without specifications, and one or more sorts of the grammar(s).

The grammars are dictionary, syntactic, word, normal form (NF), NF zero and NF non-zero. Their specifications allow for using only certain types of rules from the grammars. If no specifications occur, all rules in the grammar are used. Term, category symbol, symbol name, and value may be specified. For example:

TZ (1::CAT(N)),R

takes the NF zero grammar, chooses the rules whose left sides have the category "Noun", and displays a recognition sort of those rules.

The other sorts are form, production, and analysis. Form, production, and recognition of the dictionary grammar may also be reversed or smashed.

Action Taken:

The requests are unpacked one character per word. Any rule specifications are stored in unpacked internal representation in array IRULE. The grammars and sorts are stored in a 10 x 4 array [GRMSRT], where column 1 contains all combinations of form sorts; column 2, all combinations of production sorts; column 3, all recognition sorts; and column 4, all analysis sorts. Each entry in the columns has the lower 6 bits set to 01B, 02B, 03B, which stand for standard, smashed, and reversed, respectively. The information for each grammar in the entry is 18 bits. Bits 1 to 6 are 03B = dictionary grammar, 04B = syntactic, 05B = word, 06B = NF zero, or 07B = NF non-zero. Bits 7-12 are a pointer to the block in IRULE which contains specifications for each grammar in this particular sort. If the pointer is zero, there are no specifications and all rules are to be sorted. Bits 13 to 18 point to a block in IRSP which contains the specifications in a form used for printing sort titles.

The grammars are read, unpacked into internal representation, and written on tape 3, 4, 5, 6 or 7, depending on whether the rules are dictionary, syntactic, word, NF zero, or NF non-zero, respectively.

When a rule is used in creating a display, a line or print image is created for it by "Create Line Image" (cf. 2.1.3) and written on tape 13, 14, 15, 16 or 17 for dictionary, syntactic, word, NF zero or NF non-zero, respectively. After the line images for a grammar have been created, a flag indicating this is stored in LIST so that the action will not be repeated if the grammar is used more than once in a particular run.

Each rule in the grammar(s) being used is read and tested. If it meets the specifications (when they exist), a sort key for the particular sort is created. The data for the sort key is the line image of the rule. These are written on tape 9 which is given to SRTMRG, with common block LOSORT being 1-3=0, 4=5LTAPE9, 5=512, 6=6LTAPE10, 7-9=0. After the sort, the sort keys are thrown away. Tape 10 then contains the rules ready for output.

Use:

Input:

Control paramaters - each on separate card.

NTAPES-12-number of input tapes.
REQ-12-number of requests to be read in.
IN-80R1-REQ cards, each containing one request.

A request is in the form A+A1...A,B,B,...B, where
A = a grammar, with or without specification;
B = a sort.

A's are of the form G(T:CS:SN(V)), where
G = grammar, its values are:

D - dictionary
S - syntactic
W - word
T - NF zero and non-zero
TZ- NF zero
TN- NF non-zero.

T = term, its values are:

1 - left-side term
$2 \rightarrow N-(N-1)^{th}$ right-side term

CS = category symbol, may be any legal category symbol
SN = any legal symbol name
V = any legal symbol value.

B's may have the values:

F - Form
P - Production
R - Recognition
A - Analysis
FX- Form smashed
PX- Production smashed
RX- Recognition smashed
FR- Form Reversed
PR- Production Reversed
RR- Recognition Reversed

Smashed and reversed sorts may be used only with dictionary rules. Analysis with smashed and reversed options is equivalent to recognition smashed and reversed.

G and T <u>must</u> be present with CS and/or SN. The V option may be used only if SN is also used.

Input - Tape 1

The grammar rules which are to be displayed are input to SUBGRM. The output from SUBGRM is the input for GRMDIS2.

Output:

The number of rules occurring for each grammar and an expanded version of each request are printed first. The expanded version of T(1:ABOVE),F,P is:

```
FORM NF NON ZERO(1:ABOVE) + NF ZERO(1:ABOVE)
PRODUCTION NF NON ZERO(1:ABOVE) + NF ZERO(1:ABOVE)
```

The remainder of the output consists of the displays in the order: form sorts, production sorts, recognition sorts, and analysis sorts. The first page of each sort has a heading, the date, and expanded version. The rest of the pages have expanded version and page number.

Control Cards:

```
     [READPF(4490,SUBGRM)
     |COPYBR(INPUT,R)
     |RFL,77700.
     |NOREDUCE.
  *  |SETCORE.
     |SUBGRM(,,R,A,B)
     |RETURN(SUBGRM,PSUBGRM,R)
     |REWIND(A)
     |RETURN(TAPE1)
     [RENAME(A,TAPE1)
      READPF(0896,GRMDIS2)
      RFL,77700.
 **  [NOREDUCE.
     [REQUEST(TAPE1....)
      SETCORE.
      GRMDIS2.
```

*For data on cards which must be run through SUBGRM.
**For data on tape which is the output from SUBGRM.

2.2 Systems Programs

2.2.1 Word Choice

Word Choice is performed from right to left, based on the files which contain the antecedent (left side) WORD.

Word Choice begins by saving all FE's built upon by WORD rules. It then deletes all FE's whose antecedent is either WORD, LB, or RB. Then Word Choice procedes to go through a "flagging-unflagging" procedure wherein only the FE's which syntactic analysis will use are unflagged; all other are flagged. Word Choice unflags all the FE's directly built upon by WORD rules.

The six instructions executed by Word Choice are as follows:

1. Rejection (R) - R $T_1, T_2, \ldots, T_N$.

This operator instructs Word Choice to delete the FE's whose antecedents correspond to the right side terms $T_1, T_2, \ldots, T_N$ of the rule whose antecedent contains the D operator. All FE's building on $T_1, T_2, \ldots, T_N$ must also be deleted.

2. Insert (I) - I nSN(X) I n.m(X), where

n = term number
SN = subscript name
X = subscript value
n.m= locator.

This instructs Word Choice to insert the subscript SN with the value X in the computed left side of the FE whose antecedent corresponds to n. If the locator (n.m) format is used, the value X is added to subscript m of term n.

3. Preference (P) indicates that a particular WORD reading should have preference over all other WORD readings within the span of this WORD rule. All readings within this span that are not dominated by the P operator are deleted.

4. Superflagging (S) - S $T_1, T_2, \ldots, T_N$ guarantees that a FE stays flagged and cannot be unflagged by a WORD rule.

5. Deletion (D) - D $T_1, T_2, \ldots, T_N$. This performs the same operation as the rejection operator. However, it also unflags all FE's directly built upon by the deleted FE.

6. Setting of M conditions (M) - M $T_1, T_2, \ldots, T_N$. This instruction introduces an "M" operator into the condition columns of each file preceding and folling T_i ($1 \leq i \leq N$) (cf. p.).

2.2.2 <u>SYNT A</u> (Syntactic Analysis)

Syntactic Analysis is identical to Word Analysis [2] with the following exceptions:

1. The workspace input by SYNT A is in WORD format and needs only to be loaded. The workspace input by WORD A is in DICT format and is first converted to WORD format.

2. While WORD A can build upon any FE, SYNT A can build only upon unflagged FE's. The flagging and unflagging of FE's is done by WORD C. The information as to whether a FE is flagged or not is stored as part of the pointer of each FE.

3. While WORD A generates a list of all FE's where WORD was constructed, SYNT A generates a list of all FE's where S was constructed.

2.2.3 SVOL

The SVOL package performs the four functions: 1) subscript check, 2) value check, 3) operations, and 4) left-side construction. It is used by the majority of the analysis, synthesis, and choice programs.

2.2.3.1 SC CHECK

SC CHECK is called with three terms: 1) a term number, 2) a rule term, and 3) a workspace term. It checks to see that each subscript present in the rule term has a corresponding subscript in the workspace term. It can also check to see if either a negative condition occurs (i.e., that a subscript must not be present), or a don't-care condition (i.e., establish whether or not the workspace subscript exists, but without any decision being based upon the outcome). SC CHECK returns either true or false.

2.2.3.2 VA CHECK

VA CHECK is called with one item, a term number. For each subscript in the rule term a condition check, if necessary, is performed to determine whether or not the corresponding workspace subscript has a specified arrangement of values. This check is performed for each column. Any column that fails the condition check is eliminated from further processing. This elimination occurs for every workspace subscript containing this column. VA CHECK returns either true or false.

2.2.3.3 OPER

OPER uses as its input all the terms with which SC CHECK has been called. All the operations are first extracted from the rule terms. They are then sorted into execution order, since one operation can build upon the result of another operation. Third, starting from the beginning of the list of operations, a determination is made as to how many consecutive operations must be performed in parallel due to the fact that the workspace subscripts they build upon are slashed together. This group of operations is then resorted so that all difference operations can be done first, intersections second, and summations last. This group of operations is performed in parallel and then removed from the list. Step three is repeated until either all the operation are completed or at least one operation has returned false. OPER returns either true or false.

2.2.3.4 LS CONST

LS CONST is called with one item, a rule term. The rule term in this case is a set of instructions specifying how to build a new workspace term from all the preceding workspace terms and results of operations. LS CONST returns the new workspace term.

2.2.4 STAN A (Standard Analysis)

Standard Analysis takes as input 1) the workspace output by SYNT C, and 2) a standard grammar. It performs analysis over the sentences in the workspace using the standard grammar. The workspace is in PARALLEL format (see 2.2.5).

The analysis procedure used is similar to that employed by WORD A and SYNT A, except that STAN A performs the analysis over all standard strings of each sentence "simultaneously". This is accomplished by performing first the analysis for standard string number 1 from beginning to end in exactly the same fashion as SYNT A. During this process, each time a new FE is built the FED (file entry directory) constructed records all the standard strings for which this FE applies. In addition, for each standard string greater than 1 for which this FE applies, a flag is set to record that this FE needs no further processing for the remaining standard strings. Also, each time the analysis determines that a new file needs to be processed but that the results of such processing can only be relevant to standard strings greater than 1, all relevant information is saved until that same FE is again processed for a subsequent standard string.

After standard string 1 has been completely analyzed the process is repeated for standard string 2 from beginning to end. But this time an FE can only be in one of three states:

1. If the FE did not exist in any previous standard string, it is analyzed normally.

2. If the FE exists in a previous standard string, and if previous analysis of such a standard string has computed that no further rules can apply, this FE is not processed.

3. If the FE exists in a previous standard string, and if previous analysis of such a standard string has resulted in a portion of the analysis for this FE, the analysis resumes where it left off before.

This process is repeated for each standard string of the sentence.

2.2.5 OUT WS

OUT WS (Output Workspace) is a subroutine called by SYNT C (Syntactic Choice [3]) to rebuild the output of SYNT C into a new workspace.

The output of SYNT C is a tree which represents all of the standard strings and their component parts for all "S" readings covering the same span and for variables indicating whether or not there are "S" readings. When there are no "S" readings, a message is printed to that effect and the variables are printed. Otherwise, the tree which represents n standard strings is rebuilt into n trees, each representing one standard string.

For each set of standard strings a Sentence Directory is built. The first word in a Sentence Directory is a pointer to the first word in the next Sentence Directory, so that they are connected by a one-way list. The next two words contain pointers to the first File Entry Directory and to the last one.

File Entry Directories are also connected in a one-way list by their first words. They contain a pointer to the first word in the File Entry. The remainder of the File Entry Directory is the File Entry which already exists in the original workspace. It is merely transferred from one part of storage to another, changing any pointers to make them relative to their new positions.

When all Sentence Directories have been created, the word in the last Sentence Directory which points to the next Sentence Directory is set to zero to show that there are no more Sentence Directories. The Sentence Directories are then transferred to a tape for use by other programs. The first two words on this tape are "S" and "Next", where "Next" is a pointer to the first word after the end of the workspace. These two words constitute the first block. Then the Sentence Directories are written out in blocks of 511 words. The last block consists of one word which is "*END*".

2.2.6 TRAN CH (Validity Check of Normal Form Rules)

This program determines whether the degree associated with the normal form expression (NF expression) is equal to the sum of the non-terminated non-terminal nodes of the standard subtree analyzed by the NF expression.

TRAN CH uses as input all the NF rules and all the standard grammars (Dictionary, Word, and Syntax). All the grammars have been preprocessed and a sorted file has been made which contains only the rule number, left-side (category symbol only), number of right-side terms, and the right-side terms themselves (category symbols only) for each rule.

Given this stripped version of the grammars, each NF rule is taken and the degree for that rule computed. This computed degree must then agree with the degree coded for that rule. If there is a discrepancy, the NF rule is in error.

In addition, the first occurrence of an NF name determines the degree for all subsequent rules having the same name. All rules having that NF name must have the same degree or they are in error.

2.2.7 TRAN TC (Normal Form Tree Construction)

The sorted normal form grammar rules and pseudo rules (alpha switch rules) [4] are used as input to TRAN TC. From this input TRAN TC constructs trees representing the compiled NF rules. (The process is very similar to Dictionary TC [1] except that the right side consists of rule numbers instead of characters.)

TRAN TC reads in one entry at a time, comparing it term by term with the previous entry. Whenever a term in the new rule differs from the previous rule, a down pointer is attached to the last matched term in the previous rule to indicate the place where the new rule continues.

If the old rule is a subset of the new rule, a right pointer is attached to the old rule to indicate the continuation of the new rule.

Each entry may contain attached information consisting of 1) subscript packages, 2) left-side package (rule termination), or 3) connectors which indicate this rule involves a switch rule.

Each time a root term (rightmost term of a rule) is encountered— if it is the first time that term has been encountered (duplication numbers do not count)— a pointer to its position in the tree is placed in an index table according to the root terms' numeric value (block and word number modulo 511). To avoid large blocks containing no pointers, an index to the index table was constructed.

SECTION III

LINGUISTICS

3.1 German Syntax

3.1.1 Clause Description

3.1.1.1 Strategy for Clause Description

German clause level constituents can be permuted to a large extent. The number of clause patterns made possible by the re-arrangement of constituents is not affected by the subscript grammar. In order to reduce the number of clause patterns, it was necessary to restrict the categories which were permitted to occur on clause level. Those which we utilized are: surface subject, predicate, surface object, and adverbial.

3.1.1.1.1 Surface Subject

The surface subject appears as $\begin{matrix}\text{V NP}\\ \$\ \text{CA(N,CL)}\end{matrix}$ which signifies that the surface subject is either a noun phrase in the nominative case, or a clause. The surface subject may dominate noun phrases and subject clauses; among the latter are *dass*-clauses and verbal clauses. The German word *es* is interpreted both as an adverbial and as a noun phrase. In its anticipatory usage it is interpreted as adverbial as in

Es befanden sich drei Leute im Zimmer,

Er hatte es aufgegeben, ihn davon abzubringen.

3.1.1.1.2 The Predicate

The predicate is realized in three versions:

V PRED,
V PRED ... V PRFX
V MODAUX ... V VERBAL

PRED dominates the finite verb form of a regular verb or a modal or auxiliary which is used as a full verb; it also dominates the concatenation MODAUX VERBAL and VERBAL MODAUX.

PRFX dominates separable prefixes.

MODAUX dominates finite forms of modals and auxiliary verb forms; the German verb *lassen* is classified as being a possible modal, as in

Er liess den Mann von dem Detektiv beobachten.

The German verbs *bekommen* and *erhalten* are classified as potential auxiliaries which form the passive, as in

Er bekam (erhielt) ein Buch geschenkt.

VERBAL dominates non-finite forms of full verbs, zu-infinitives, and concatenations of non-finite verb forms and nonfinite modals or auxiliaries.

3.1.1.1.3 Objects

Objects are realized as $\genfrac{}{}{0pt}{}{V\ NP}{\$\ CA}$. This symbol dominates noun phrases and object clauses (cf. 3.1.1.1).

3.1.1.1.4 Adverbials

Adverbials appear as $\genfrac{}{}{0pt}{}{V\ ADV}{?\ CA}$. They dominate one-word adverbs, prepositional phrases, prepositional objects, subordinate clauses, and noun phrases which function as adverbials of extension in space and time, as *vierzehn Jahre* in

Er arbeitete vierzehn Jahre.

3.1.1.2 Clause Patterns

Our linguistic description of three chapters of a work on aeronautics, *Raketenantriebe: Ihre Entwicklung, Anwendung und Zukunft* [5], resulted in approximately 1100 different clause patterns for the 150 pages of text. Rather than describe only the occurring patterns, we decided to generate the complete set of clause patterns systematically. This was done in order to determine the extent to which coverage would be sufficient for the analysis of these 150 pages as well as for the remainder of this text or any other text.

The following basic patterns were assumed.

1. S-P
2. S-P-A
3. S-P-A-A
4. S-P-A-A-A
5. S-P-A-A-A-A

6. S-P-0
7. S-P-0-A
8. S-P-0-A-A
9. S-P-0-A-A-A
10. S-P-0-A-A-A-A

11. S-P-0-0
12. S-P-0-0-A
13. S-P-0-0-A-A
14. S-P-0-0-A-A-A
15. S-P-0-0-A-A-A-A

16a. P
16b. P-A
17. P-A-A
18. P-A-A-A
19. P-A-A-A-A

20. P-0-A
21. P-0-A-A
22. P-0-A-A-A
23. P-0-A-A-A-A

where S = subject, P = predicate, A = adverbial, 0 = object. The patterns 16a-23 can occur only in the passive voice or imperative mood. Examples:

Wurde gearbeitet? (P)

Es wurde gearbeitet, wurde daran gearbeitet? (P-A)

The number of variants for the 24 basic patterns was almost tripled by the fact that the predicate can occur in the three different versions, as described above under 3.1.1.2. Moreover, all terms except P were allowed to occur in any position. (The restrictions on the position in which P may occur are given below.) The decision to permit free permutation of all non-predicate clause constituents may seem excessive. However, repeated inspection of doubtful arrangements has shown that in all instances perfectly well-formed German sentences could be so constructed.

3.1.1.3 Restriction of Predicate Position

The predicate may occur only in four positions: as the first, second, penultimate, or ultimate term of a rule consequent. Examples:

Bestand er darauf? Er bestand immer darauf. (Weil) er immer darauf bestand, Recht zu haben.

If the predicate consists of two elements, the finite verb part is restricted to positions one or two, the non-finite to the penultimate or ultimate. Note that if PRED, VERBAL, or PRFX occurs in penultimate position (where penultimate is not the first or second position), the ultimate position must contain a clause.

3.1.1.4 Coding of Clause Patterns

The projected task of clause description is represented in the following chart.

Number of Possible German Syntactic Pattern Variants

Patterns with 2 right-hand terms:

1.	CLS←S P		5 rules
16b.	CLS←P A	Passive only	3 rules
		Sum:	8

Patterns with 3 terms:

2.	CLS←S P A		17 rules
6.	CLS←S P O		17 rules
17.	CLS←P A A }	Passive only	6 rules
20.	CLS←P O A }		
		Sum:	52

Patterns with 4 terms:

3.	CLS←S P A A		31 rules
7.	CLS←S P O A		62 rules
11.	CLS←S P O O		31 rules
18.	CLS←P A A A }	Passive only	24 rules
21.	CLS←P O A A }		
		Sum:	156

Patterns with 5 terms:

4.	CLS←S P A A A		43 rules
8.	CLS←S P O A A		129 rules
12.	CLS←S P O O A		129 rules
19.	CLS←P A A A A }	Passive only	32 rules
22.	CLS←P O A A A }		
		Sum:	341

Patterns with 6 terms:

5.	CLS←S P A A A A		56 rules
9.	CLS←S P O A A A		220 rules
13.	CLS←S P O O A A	}Passive	276 rules
23.	CLS←P O A A A A	only	40 rules
		Sum:	592

Patterns with 7 terms:

10.	CLS←S P O A A A A		335 rules
14.	CLS←S P O O A A A		670 rules
		Sum:	1005

Patterns with 8 terms:

15.	CLS←S P O O A A A A	1185 rules

<u>Totals</u>:

Patterns with 1 term	1 rule
Patterns with 2 terms	8 rules
Pattners with 3 terms	52 rules
Patterns with 4 terms	156 rules
Patterns with 5 terms	341 rules
Patterns with 6 terms	592 rules
Patterns with 7 terms	1005 rules
Patterns with 8 terms	1185 rules
Total:	3340

Work on the generation of those variants has resulted in 315 rules coded so far.

In order to reduce the number of necessary clause rules to more readily managable proportions, a change of the analysis algorithms to operate with set theoretical rules is envisioned. The number of clause rules will be further reduced by the introduction of optional terms.

3.1.1.5 <u>Clause Types</u>

No decisions as to the type of clause—declarative, interrogative, relative, etc.—are made in the actual analysis part of a clause rule. Thus, the clause rule 186 (cf. below) analyzes the underlined strings in the following examples:

Der Mann, <u>der ihn damals besuchte</u>,

der fragte, <u>wer ihn damals angezeigt habe</u>,

er sagte es, weil <u>er ihn nicht leiden konnte</u>.

The information necessary to determine the actual clause type is contained in the antecedent of the clause rule. Information pertaining to interrogative and relative clauses is contained under the subscript FM. Subordinate clauses are recognized by their predicate position (penultimate or ultimate) or by the subjunctive mood of the verb. Rules 219 and 220 below thus analyze a clause as a noun phrase; rule 222 determines the occurrence of a relative clause; rules 218, 228, and 229 analyze a subordinate clause as an adverb and determine the position in which it can occur in its matrix clause.

C 218	V ADV	V CONJ	V CLS	V PNCT	
	+ POS(IN)	$ KT(S)	$ FL	$ TY(COM)	
	+ EX				
C 219	V NP	V PNCT	V CONJ	V CLS	
	+ CA(CL,P+ CL)	$ TY(COM)	$ WD(TH)	$ FL	
	+ TY(TH)				
	+ POS(FL)				
	+ EX				
C 220	V NP	V PNCT	V CONJ	V CLS	V PNCT
	+ CA(CL,P+ CL)	$ TY(COM)	$ WD(TH)	$ FL	$ TY(COM)
	+ TY(TH)				
	+ POS(MED)				
	+ EX				
C 222	V CLSREL	V CLS			
	- 2.3	$ G			
	= 2	$ N			
		$ FL			
		> WD			
C 228	V ADV	V PNCT	V CONJ	V CLS	V PNCT
	+ EX	$ TY(COM)	$ CJ(S)	$ FL	$ TY(COM)
	+ POS(MED)		$ KT(S)		
	1				
	S 3-4				
C 229	V ADV	V PNCT	V CONJ	V CLS	
	+ EX	$ TY(COM)	$ CJ(S)	$ FL	
	+ POS(FL)		$ KT(S)		
	1				
	S 3-4				

3.1.1.6 Information in Clause Rules

The decisions that need to be made in the syntactic part of a clause rule may be described by discussing the following rule.

TORM SYNTACTIC

```
C 10A     V CLS        V AP          V WP           V ANV          V PRED      N LEFT     S RIGHT     N AJE        N VOICE
          S 4.1FL      O PO(N-CL)    2 POS(NCO)     O ES           S FL                               S 1.17W      O VC(A)
          S 4.2PX      D POS(1W)     S PA           2 POS(WFN)     2 PX(LO)                           S 1.6HM
          S 4.5TH      D FN          - 3.2W0.8      2 CA           S P3
          S 4.8HO      O PS                         -24.3.9.3      S WU
          SP2.5        O WU                                        S TH
          S.3.3FOI     . P.4.5.3                                   S HO
          S 2.5WD      . P.5.5.4                                   2 4E
          S 7.98       D VD                                        S FO
          S 2.10H      D A                                         P
          S 2.3FN      D W
          S*3.3VO      P
          S.5.7AX
          S.4.4A1

          1
          R CT(S)

          2
          N SHECAV-I
            (4)

          3
          C 1.20.P/S

          4
          C 7/11.P/C
          R PC(VENC)
            5.2.3.4)
          S 3*(VC)

          5
          C 1.21.P/T

          6
          C 7/11.P/C
          R AC(VSNC)
            5.2.3.4)
          S 9(VC)

          7
          C 1.22.P/0

          8
          C V/11.P/C
          R CC(SOC(2
            .3.4)
          S 9(P)
          S 9(VC)

          9
          C 1.9W1.4.
            P/C

          10
          C P/C
          P LC(VSNC)
            5.2.3.4)

          11

          R E0-312.5
            .4)

          12
          X DS
          S 6(L)
          S 7(R)
          S 8(P)

          13
          S L-5-N-90
            -VC-F-N-
            P-LC-0-N
            2-03-1-2
            -3-4-0

                     N VOICE
                     O VC(P)
```

BEST COPY AVAILABLE

The consequent part of rule C 186 (terms 2-5) analyzes any string consisting of a surface subject, followed by a surface object, followed by an adverb, followed by a predicate. The subject must either be in the nominative case or it must be a clause. The subject must also agree in person and number with the predicate.

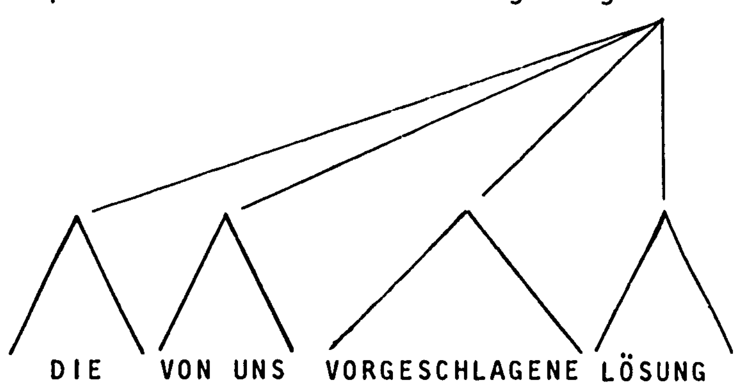
The predicate may be composed of an auxiliary and a non-finite verb form or it may be a finite verb. It must not occur with a prefix [PX(LA)], and it must govern the case of the surface object and may govern the case of the surface "adverb" if it is a prepositional object.

The antecedent of rule C 186 stores the tense and mood information of the predicate as well as information about the type of auxiliary which occurs if the predicate consists of a finite verb and a non-finite verb. The antecedent also stores the information that the predicate is in final position [FL]. It stores the case of the governed object [FOI] and the case information, if any, of other objects which the verb governs obligatorily [FO]. It stores the preposition of the "adverb" if it is a prepositional object [AI].

If the subject is a pronoun, the antecedent stores information as to whether it is an interrogative pronoun [FM], or a relative pronoun [WD,G,N].

3.1.2 <u>Noun Phrases</u>

Verbal participles can function as adjectives. Thus the variety of lause patterns would theoretically be repeated for noun phrases if strings like *die von uns vorgeschlagene Lösung* were interpreted as in the following diagram:



Note that *von uns* is the deep subject of the verb *vorschlagen*, and *Lösung* its object.

Noun phrases are described as consisting of up to three terms: determiner, adjective phrase, and noun. The rule system on p. I.2 was designed to guarantee agreement between adjectivally used participles and their modificands. In the case of present participles, this agreement corresponds to that between the underlying verb and its subject. In the case of the past participle, it is between the underlying verb and its object. The solution was facilitated by the fact that such adjectival verb forms within a noun phrase do not occur with sentence complements.

3.1.3 German Determiners and Pronouns

German determiners are marked for gender, number, case, and inflection (strong or weak), as described earlier in [3].

In addition to gender, number, and case markers, pronouns have a subscript FM (form), whose values identify them as P+DEM (demonstrative pronoun), P+PERS (personal pronoun), P+REL (relative pronoun), P+REF (reflexive pronoun), P+INT (interrogative pronoun), P+IND (indefinite pronoun), P+POSS (possessive pronoun), or P+REC (reciprocal pronoun). All those German dictionary items which may function as either determiner or pronoun, depending on their environment, were coded only once, with a complex label which contains their features as determiners and their features as pronouns. (They are identified as possible determiners by the value DET under the subscript FM.) This prevents multiple analyses of such items regardless of their environment, a considerable savings because of their frequency of occurrence in actual texts. Examples are shown on the following page taken from the German LRC dictionary.

```
C 198      V DET          . DENSELHEN
D 4        . UD(H#M.F     P
             .H)/
           . NU(S#P)/
           . CA(A#D)
           . IN(S)
           . FM(DET.P
             .EN)
           . TY(MH.AL
             .PL.IN.N
             T.AN)
           -

C 149      V DET          . DEN
D 1        . UD(N#F#N     P
             .F.N)/
           . NU(S#S#P
             )/
           . CA(N#U.D
             #G)
           . IN(S)
           . FM(DET.P
             .EN.PEN
             EL)
           . U(N#P)/
           . C(N#D)
           . N(S)
           . TY(MH.AL
             .PL.IN.N
             T.AN)
           -
           I 1.5

C 197      V DET          . DENJENIGE
D 1        . UD(N)        P
           . NU(S)
           . CA(N)
           . IN(S)
           . FM(DET.P
             .EN)
           . TY(MH.AL
             .PL.IN.N
             T.AN)
           T 1.5
           -
```

Rule $\begin{smallmatrix} C & 149 \\ D & 1 \end{smallmatrix}$ classifies the item *der* as determiner, relative pronoun, and demonstrative pronoun.

Relative pronouns were given the subscripts G, C, and N in addition to GD, CA, and NU. The latter set of features is for agreement with the following nominal in relative noun phrases:

> *Es geschah in 1963, zu welcher Zeit derartiges noch ganz unerwartet war.*

The features G and N must agree with the gender and number of the preceding nominal which is being modified by the relative clause. C (case) must contain the case governed by the verb:

> *die Explosion, deren man sich noch heute erinnert.*

The subscript TY (semantic type) is added to all pronoun entries. It reflects the semantic type of noun each pronoun may represent. Thus, the German relative pronoun *was* is assigned the feature TY(AB) for "abstract"; the indefinite pronoun *jemand*, the feature TY(HU) for "human"; the pronoun *der*, all possible main semantic classes: TY(HU,AL,PL,IN,NT,AB).

Personal pronouns are also given the subscript PS (person) with the possible values 1, 2, and 3 for 1st, 2nd and 3rd person.

Some pronouns or determiners which must be recognizable as a specific lexical item are marked by the subscript WD (word) with an identifying abbreviation as value. For example, only the item *was* may be used as a relative pronoun modifying a clause rather than a noun or noun phrase. For this reason, it contains the feature WD(W).

The rules which analyze pronoun as NP's assign the gender, number, case, and semantic type features of the pronoun in the workspace to the NP. For personal pronouns, the PS feature is also assigned to the NP; for all other pronouns, the new feature PS(3) marks the NP as 3rd person.

3.2 Choice Rules

3.2.1 Function

It is the function of the choice rules associated with a particular syntactic rule:

to determine the deep structure of the string interpreted by the rule, based on the semo-syntactic features associated with the clause constituents, and

to generate that deep structure, called standard string.

This is performed by permuting the clause level constituents, by adding new terminal symbols (dummy terms or standard terminals), and by deleting certain surface terminals, such as prefixes or the reflexive pronoun in cases of actual reflexive verbs. (*Sich beeilen* vs. *sich waschen*; **ich beeile ihn*, but *ich wasche ihn*.)

An additional function of the choice rules is the elimination of forced ambiguous readings.

3.2.2 Determination of the Deep Structure

The deep structure of a given sentence is determined in several phases.

3.2.2.1 Determination of Clause Type

We distinguish between active clauses, passive clauses, copula clauses, and *lassen* clauses. *Lassen* clauses are further divided into *lassen* clauses with an embedded active clause, and such with an embedded passive clause. Example:

(Active) *Er liess den Jungen den Hund schlagen.*

(Passive) *Er liess den Hund von dem Jungen schlagen.*

The clause type is determined by evaluating the result of the intersection between the values of the subscript TY of the constituent MODAUX and the values of the subscript AX of the non-finite verb part. The actual decisions are made by choice rule V CT which is called with the information contained in the rule antecedent and the full verb (PRED or VERBAL).

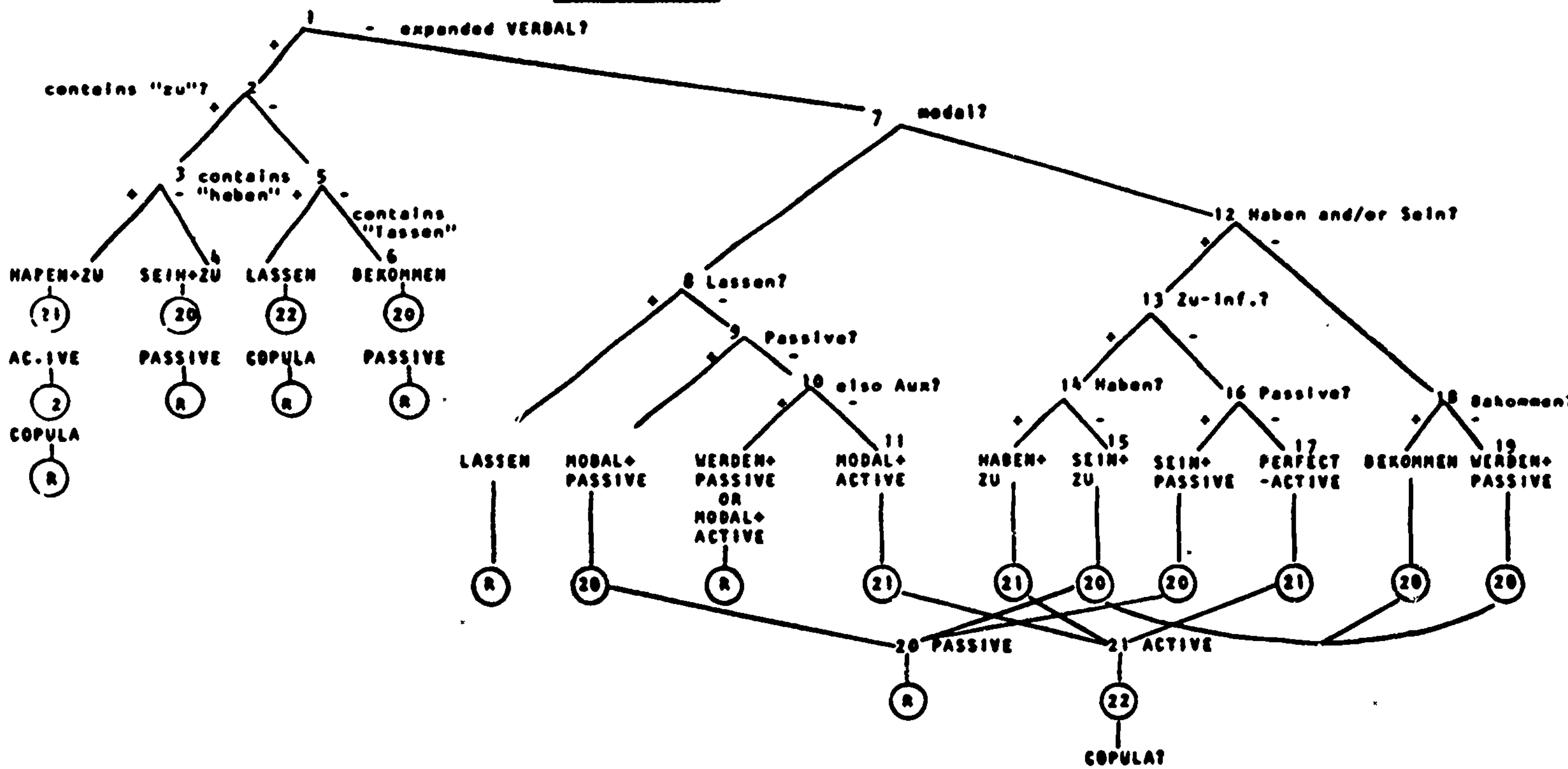
PRODUCTION SYNTACTIC

```
        C 1        V CT         V̄ CT

                   1            1̄
                   C F/7        * ĊT(-M)

                   2            2̄
                   C F/5        * ĊT(MI)

                   3            3̇
                   C T/21       * ĊT(M)
                   X NHABEN.Z
                     U          5̄
                                * ĊT(L)
                   4
                   C .3.T/20    6̇
                   X NSEIN.ZU   * VC(P)

                   5            1̄3
                   C T/22       * ZU
                   X NLASSEN
                                1̄6
                   6            * V̇C(A)
                   C .5.T/20
                   X NBEKOMME   2̄2
                     N          * ṪY(C)

                   7
                   C F/12
                   S AX(M)

                   8
                   C T/22
                   S AX(M.L)
                   X NLASSEN

                   9
                   C T/20
                   X NMODAL.P
                     ASSIVE

                   10
                   C T/R
                   S AX(A)
                   X NWERDEN.
                     PASSIVE-
                     OR-MODAL
                     .ACTIVE

                   11
                   C .10.T/21
                   X NMODAL.A
                     CTIVE

                   12
                   C F/18
                   S AX(MZVSZ
                     VA.MVA.S
                     )

                   13



                   C F/16
                   S AX(M.VSZ
                     )
```

```
C F/16
S AX(HZvSZ
  )

14
C T/21
S AX(HZ)
X NHABEN+Z
  U

15
C +14,T/20
X NSEIN+ZU

16
C T/20
S AX(A+S)
X NSEIN+PA
  SSIVE

17
C +16,T/21
X NPERFECT
  -ACTIVE

18
C T/20
S AX(A+W+R
  )
X NBEKOMME
  N

19
C +18,T/20
X NWERDEN+
  PASSIVE

20
C 4v6v9v15
  v16v18v1
  9,T/R
X NPASSIVE

21
C 3v11v14v
  17
X NACTIVE

22
C 5v8v11v1
  7,T/R,F/
  R
X NCOPULA
```

The decisions made in this rule are represented by the following graph:

Type of Clause



The numbers in the graph correspond to the numbers in the choice rule.

The decisions made by this choice rule are not final; the actual determination of the clause type is dependent on the relations between the predicate and its complements. Thus, *addieren* (*to add*), will first be interpreted as active voice in a sentence such as *die Zahlen addieren sich zu hundert*. The verb complement choice rule V AC:VSOA (for active complete: verb, subject, object, adverb) will, however, assign to this clause the interpretation "passive voice". This permits both the translation *the numbers are added up to a hundred* and *the numbers add up to a hundred*. Similarly, *es wird getanzt*, which superficially looks like a passive sentence, will be interpreted as an active sentence with a deleted agent, which permits the translations *they danced* or *people danced*.

3.2.2.2 Determination of Adverbials

The choice rule called V SPECAV (special adverb) determines whether an adverb is the negation *nicht* (*not*), or an adverb of the type *gern, lieber, weiter* which function as deep predicates. The negation is moved directly behind the subject to facilitate the generation of the English output; it could, however, be put in front or behind the actual clause to indicate its operator status. This difference in treatment would not have any effect on the translation. The special adverbs of the type *gern, lieber* are moved into the predicate position. The surface predicate and its object complements are treated as the clause complement of the predicate represented by the surface adverb.

Adverbials which dominate prepositional phrases undergo further checking in the verb complement choice rules. There, we determine whether or not the adverb is a prepositional object of the predicate. In the final assignment statement, each adverb is assigned a numerical value (cf. choice set number 12 in rule C 186 above). These numerical values are deleted if they compete with an alphameric name such as N, SP, or O, O_2, O_3.

3.2.2.3 Verb Complement Rules

We distinguish four types of verb complement rules: those which contain a passive predicate, an active predicate, a copula, or a form of *lassen*. The corresponding choice rules begin with the letters PC, AC, CC, and LC. For each set of these choice rules there exist as many alternants as there are basic sentence patterns (cf. the patterns in 3.1.1.2 above). The functions of verb complement rules are basically four-fold:

a. they determine whether the subject and object of the verbs agree with the verb in syntactic surface appearance and semantic type;

b. if these tests fail, we test for the occurrence of a different clause type (remember *addieren sich* and *wird getanzt* above);

c. after that we test for the occurrence of a lexical collocation. This test is executed superficially only, by checking whether the verb and the constituent in question agree in their values of LC (for more details, cf. 3.4.2 below);

d. if all of these tests fail, the main rule which called the choice rule is rejected.

3.2.2.4 Superscript Assignment

The clause constituents are finally connected in the sequence represented by the final superscript assignment (statement number 13 in rule C 186, p. III.22). The sequence L-S-N-SP-VC-F-M-P-LC-$0-0_2-0_3$-1-2-3-4-R stands for "left boundary, deep subject, negation, special adverb, voice information, tense information, auxiliary or modal, predicate, lexical collocation, first deep object, second deep object, third deep object, first adverb, second adverb, third adverb, fourth adverb, right boundary". Of these, only left boundary, subject, voice, tense, predicate, and right boundary are obligatory. If a constituent is assigned more than one alphabetic superscript name, the Cartesian product of permissible standard strings will be generated, with the provision,

a. that no two identical names may occur in the same standard string, and

b. that identical standard sequences which were derived by means of assigning different names to the same constituent are conflated to one standard string.

Currently, choice rules can only be called from a main rule, but we plan to extend the algorithms' capability to that of calling a choice rule from a choice rule. However, a restriction will be imposed, in that a choice rule which had been called by another choice rule may not call a third choice rule.

3.3 Standard Grammar

3.3.1 Economy of Standard Descriptor

The number of standard grammar rules is fairly small. There are three reasons for this:

a. the constituents occur in their deep structure order,

b. relevant boundary information is retained by means of dummy terminals, and

c. surface structure, which is identical to deep structure, is retained.

Surface structure is destroyed in those cases where each node labeled clause, or dominating a clause, or dominating a standard expanded adjective, is destroyed. Adjectives are expanded if they concatenate with an object complement or a sentential adverbial.

Clause rules are destroyed because each clause rule introduces at least three dummy terms and these dummy terms must be incorporated into the standard structural description.

Strings containing an expanded adjective phrase are rearranged to represent standard order.

3.3.2 Standard Clause Patterns

Concatenations of the predicate and its complements are interpreted by the symbol KERN. There are exactly four types of structures dominated by KERN.

1. Structures consisting of subject and predicate and no object.

2. Structures consisting of subject and predicate and exactly one object.

3. Structures with subject, predicate, and two objects.

4. Structures and predicate and three objects.

These four structures are represented by the following standard rules.

```
C ?       V AUX       V AUX       V FIN
          $ 3.1                   $ TY(A)
          A 2                     $ WD

C 40141   V CLS       V LEFT      V KERN      V RIGHT

C 4       V FIN       V FIN       V FIN
          = 2         $ TY(A)     $ WD(W)
                      $ WD(S)     $ AX(S)
                                  $ TY(A)

C 40139   V KERN      V ARG       V PRED      V ARG       V ARG       V PARG      V VB
          $*2.2S      STTY        $ TO(0.02.  $ OR(0)     $ OR(02)    $ OR(03)
          $*6.5F      . 2.1.3.2    03)        STTY        STTY        $ CA
          $.6.503                 STTS        - 4.2.3.4   - 5.2.4.3   $TTY
          $.5.302                 $ FO        - 4.1.3.1   - 5.1.4.4   - 6.2.3.3
          $.4.30                  STTO                                - 6.3.5.3
                                                                      - 6.1.5.4
          1
          A 2TY.3TS(
            1S)
          A 3TO.4TY(
            10)
          A 3TO.5TY(
            102)
          A 3TO.6TY(
            103)

C 40136   V KERN      V ARG       V PRED      V ARG       V ARG       V VB
          $*2.2S      STTY        $ TO(0.02.  STTY        STTY
          $*5.2F      . 2.1.3.2    *03)       - 4.1.3.3   - 5.1.4.2
          $.4.20                  STTS        $ OR(0)     $ OR(02)
          $.5.202                 STTO        - 4.3.3.1   - 5.3.4.4
          1
          A 2TY.3TS(
            1S)
          A 3TO.4TY(
            10)
          A 3TO.5TY(
            102)
```

```
C 40137   V KERN      V ARG       V PRED       V ARG       V PARG      V VB
          $*2.2S      $TTY        $ TO(0.02.   $TTY        $ OR(02)
          $*5.4F      . 2.1.3.2     *03)       - 4.1.3.4   $ CA
          $.4.20                  $TTS         $ OR(0)     $TTY
          $.5.402                 $ FO         - 4.3.3.1   - 5.3.4.2
                                  $TTO                     - 5.2.3.3
                                                           - 5.1.4.4
          1
          A 2TY.3TS(
            1S)
          A 3TO.4TY(
            1O)
          A 3TO.5TY(
            1O2)

C 40134   V KERN      V ARG       V PRED       V ARG       V VB
          $*2.2S      $TTY        $ TO(0A*02   $TTY
          $.4.20      . 2.1.3.2     A*03)      - 4.1.3.3
                                  $TTS
          1                       $TTO
          A 2TY.3TS(
            1S)
          A 3TO.4TY(
            1O)

C 40136   V KERN      V ARG       V PRED       V PARG      V PARG      V VB
          $*2.2S      $TTY        $ TO(0.02.   $ OR(0)     $ OR(02)
          $*5.6F      . 2.1.3.2     *03)       $ CA        $ CA
          $.5.602                 $TTS         $TTY        $TTY
          $.4.60                  $ FO         - 4.2.3.3   - 5.2.4.4
                                  $TTO         - 4.1.3.1   - 5.1.4.5
                                               - 4.3.3.4   - 5.3.4.6
          1
          A 2TY.3TS(
            1S)
          A 3TO.4TY(
            1O)
          A 3TO.5TY(
            1O2)

C 40135   V KERN      V ARG       V PRED       V PARG      V VB
          $*2.2S      $TTY        $ TO(0A*02   $TCA(P)
          $.4.40      . 2.1.3.2     A*03)      $TTY
                                  $TTS         - 4.1.3.3
          1                       $ FO         - 4.2.3.4
          A 2TY.3TS(              $TTO
            1S)
          A 3TO.4TY(
            1O)

C 40133   V KERN      V ARG       V PRED       V VB
          $*2.2S      $TTY        $ TO(LA)
                      . 2.1.3.2   $TTS
          1
          A 2TY.3TS(
            1S)
```

```
C 40140    V KERN      V KERN      V ADV

C 2        V MODAL     V AUX       V FIN
           = 3                     $ TY(M)
           ^ 2                     $ WD

C 3        V MODAL     V MODAL     V FIN
           ^ 2                     $ TY(A)

C 8        V MODAL     V MODAL     V FIN
           ^ 2                     $ TY(M)
                                   $ WD(*W)

C 6        V VB        V AUX       V V
           ^ 3

C 5        V VB        V MODAL     V V
           ^ 3
```

The symbol VB stands for verb phrase boundary. It is introduced in the verb complementation choice rules.

Sentence adverbials concatenate in binary rules with the symbol KERN recursively.

As can be seen from rules C 40133 and C 40139 above, standard analysis is followed by standard choice. There are only two types of instructions executed in standard choice: assignment statements, which help to select the proper translation equivalents, and superscript assignment statements, which change the order of the standard terms to the universal order if the two should be different.

3.4 Lexicography

During this reporting period, the lexical data base necessary for the analysis and translation of the text *Raketenantriebe: Ihre Entwicklung, Anwendung und Zukunft* was completed and revised. The lexical data base consists of four dictionaries— a German monolingual dictionary, which lists the syntactic and semo-syntactic features of the German word stems in the form of subscripts and values; a German normal form dictionary, which establishes meaning (or translation equivalence) classes for the lexical entries in the German monolingual dictionary; an English monolingual dictionary, which assigns features to English word stems; and an English normal form dictionary, which establishes translation equivalence classes for the English dictionary items.

3.4.1 Verb Entries

The general feature system for verbs, nouns, and adjectives is described in [1]. This system was used in the coding of monolingual dictionary rules for German and English verbs, with the following minor modifications:

a. FS (syntactic form of subject required by the verb) is always shown as either N (nominative NP) or CL (clause);

b. the subscript TS (semantic type of subject) lists, for those verbs which allow such subjects, the values TH (*that*-clause), MI (marked infinitive), ICL (interrogative clause), FT (*for-to* complement, English only), or IT (impersonal subject *es* or *it*), in addition to semantic noun classes;

c. the earlier subscript DS (deep subject) was replaced in the dictionary rules by IS (interpretation of subject). Its possible values are S (subject, where surface and deep subject are identical) and O (direct object, where the surface subject represents the deep object);

d. analogous to the subscript IS, the new subscript IO (interpretation of object) was introduced, with the possible values O (direct object), O2 (indirect or "second object), O3 (third object), and S (subject; for those verbs whose surface object represents the deep subject as in *dieser Versuch gelang ihm = he succeeded in this attempt*);

e. the subscript OB was changed to FO (form of object). Its possible values remain N, G, D, A (for the cases in German), O (for NP objects in English), and all prepositions which can be used in prepositional objects of verbs. An additional value is CL (clause). The values TH (*that*-clause), MI (marked infinitive), etc. (cf. b. above) are also listed under TO (semantic type of object), in addition to the semantic noun classes. Double objects are indicated by "." under FO, TO and IO. A "+" is used to combine two values into one, as in AUF1+CL, which stands for "the object consists of the preposition *auf* followed by a clause", as in *darauf achten, dass...*

In addition to these semo-syntactic and syntactic features, each verb stem was assigned the subscript CL (paradigmatic class) with the number of the specific inflectional class it belongs to, and the subscript PX (separable prefix) with the values LA (lambda = no prefix) or AB, AN, AUF, etc.

For examples of German and English monolingual dicationary rules for verbs, cf. [3], pp. II-29 and II-30. The "≠" sign in the last two rules on both pages separates "columns" of features. The value shown preceding the ≠ in one subscript forms a feature packet with all other values preceding the ≠ in other subscripts. Another feature packet is formed by the values following the ≠. Examples of the normal form rules for the German verb *ändern* (with the prefix *ab* but without a separable prefix) and its English translation equivalents *change* and *modify* are marked on the three following pages taken from the German and English normal form dictionaries (pp. III.39 and III.40 , respectively).

RECOGNITION TRANSFER ZERO(1::CAT(V))

```
C 23         V COMPLETE   C 4000
             A CAT(V)     $ PX(LA)
             N TH(ABSOL
               VIEREN)

C 24         V ADD        C 4001
             + FO(O.TO)   $ PX(LA)
             A CAT(V)
             N TH(ADDIE
               REN)

C 25         V CHANGE     C 4002
             + FO(LA)     $ PX(LA)
             A CAT(V)     $ TO(R)
             N TH(AENDE
               RN)

C 392        V CHANGE     C 4002
             + FO(O)      $ PX(LA)
             A CAT(V)     $ TO(*R)
             N TH(AENDE
               RN)

C 2          V MODIFY     C 4002
             A CAT(V)     $ PX(AB)
             N TH(ABAEN
               DERN)

C 26         V EXERT      C 4004
             A CAT(V)     $ PX(LA)
             N TH(AEUSS
               ERN)

C 27         V ACT        C 4005
             + FO(AS)     $ PX(LA)
             A CAT(V)
             N TH(AM  N
               )

C 87         V DEVELOP    C 4006
             + FO(O)      $ PX(AUS)
             A CAT(V)
             N TH(AUSAR
               BEITEN)

C 234        V PERFECT    C 4006
             A CAT(V)     $ PX(DURCH
             N TH(DURCH     )
               ARBEITEN
               )
```

RECOGNITION TRANSFER ZERO(1::CAT(V))

C 1	V BURN+OFF	C 4000
	+ FO(0)	$ FO(0)
	A CAT(V)	$ PX(OFF)
C 3	V MODIFY	C 4001
	A CAT(V)	
C 5	V FIRE	C 4004
	+ FO(0)	$ FO(0)
	A CAT(V)	
C 620	V FIRE	C 4004
	+ FO(ON,UP ON)	$ FO(ON,UP ON)
	A CAT(V)	
C 6	V UNCOVER	C 4005
	A CAT(V)	
C 8	V DELIVER	C 4007
	$ 2.1IO	$ IO
	A CAT(V)	
C 41	V SUCK+IN	C 4008
	A CAT(V)	$ PX(IN)
C 9	V SUCK+OFF	C 4008
	A CAT(V)	$ PX(OFF)
C 10	V SEPARATE	C 4009
	A CAT(V)	
C 11	V SEAL+OFF	C 4010
	$ 2.1IO	$ IO
	A CAT(V)	$ PX(OFF)
C 12	V TRUNCATE	C 4011
	A CAT(V)	
C 487	V LAUNCH	C 4012
	$ 2.1IO	$ IO
	A CAT(V)	
C 15	V OMIT	C 4014
	A CAT(V)	
C 16	V DISCHARG E	C 4015
	A CAT(V)	

RECOGNITION TRANSFER ZERO(1::CAT(V))

```
C 407        V ENCOURAG    C 4398
               E           S IO
             S 2.1
             A CAT(V)

C 410        V FILL        C 4401
             + FO(O,WIT    S PX(LA)
               H)          S FO(O.WIT
             A CAT(V)        H)

C 411        V FILL        C 4401
             + FO(O)       S PX(LA)
             A CAT(V)      S FO(O.*X)

C 415        V FOUND       C 4405
             A CAT(V)

C 52         V HOLD        C 4406
             + FO(FOR)     S FO(FOR)
             A CAT(V)

C 387        V HOLD        C 4406
             + FO(LA)      S PX(LA)
             A CAT(V)      S FO(LA)

C 416        V HOLD        C 4406
             + FO(O,TO)    S PX(LA)
             A CAT(V)      S FO(O.TO)

C 417        V HOLD        C 4406
             + FO(O)       S PX(LA)
             A CAT(V)      S FO(O.*X)

C 418        V HANDLE      C 4407
             A CAT(V)

C 23         V CHANGE      C 4410
             + FO(LA)      S FO(LA)
             A CAT(V)

C 388        V CHANGE      C 4410
             + FO(O)       S FO(O.*X)
             A CAT(V)

C 420        V CHANGE      C 4410
             + FO(TO)      S FO(*X.TO
             A CAT(V)        )

C 786        V CHANGE      C 4410
             + FO(O.INT    S FO(O.INT
               O)            O)
             A CAT(V)
```

In these rules, the 1st column is a unique number identifying each rule. The second column contains the class name and certain information to facilitate rule inspection by the human users. For their convenience, the translation equivalence class names used are the English canonical forms, together with subscripts introduced by "+" on the left side, where necessary. For example, the German verb *ändern* with the prefix *ab* and the English verb *modify* are both members of the class MODIFY. No distinguishing subscripts are necessary. The German verb *sich ändern* is in the class $\mathrm{CHANGE}_{\mathrm{FO(LA)}}$ to insure its translation into the intransitive form of the English verb *change*. The German verb *etwas oder jemanden ändern* is in the class $\mathrm{CHANGE}_{\mathrm{FO(O)}}$, as is the transitive form of the English verb *change*.

The additional subscripts are only for convenience and do not distinguish class names: CAT(V) stands for "category symbol 'verb'" to indicate that the classified dictionary item is, e.g., the verb and not the noun *change*. In the German normal form rules, an additional subscript, e.g., TM(AENDERN), indicates the particular German verb covered by this rule.

The third column contains the right-hand sides of these normal form rules. Here, the first line contains the number uniquely identifying a dictionary rule in the German or English monolingual dictionary, respectively. Where necessary, this number is followed in the second and subsequent lines by subscript conditions which list the restrictions on the application of the normal form rule. For example, C 4002 identifies the German dictionary entry for *ändern*. The normal form rule C 2 (page III-39) applies to this verb only if it is used with the prefix *ab* [this condition is expressed by $ PX(AB)] and guarantees translation into *modify*. Rule C 25 applies if the verb *ändern* is used reflexively [$ TO(R)] and without prefix [$ PX(LA)]. Rule C 392 applies if the same verb is used without prefix and not reflexively [$ PX(LA) $ TO(*R)].

3.4.2 Dictionary Entries for Verbal Lexical Collocations

3.4.2.1 Entries Without Internal Variables

We refer to those verbs as "verbal lexical collocations" which consist of a verb and a noun phrase, a prepositional phrase, a non-finite verb form, an adjective, or an adverb, e.g., *erfolgen = take place; ins Gewicht fallen = be important*. Since such lexial collocations have features and meanings which may not be derived from their individual components, they must be treated as lexical phrases (cf. [2], pp. 13 ff.). However, since their components frequently occur discontunously and in various sequences, they must be handled differently from normal verbs. For this purpose, the subscript LC (lexical collocation) was added to the dictionary entries of those nouns and verbs which may occur as components of lexical collocations. Surface analysis refers to these subscripts and guarantees that the items occur contiguously and in a predefined order in the so-called "standard string" (cf. III.3.2.2.4), which is generated after surface analysis.

For the actual analysis and translation of lexical collocations, standard dictionary rules were coded which are applied to the standard strings.

The constituents of standard strings are the dictionary readings of the underlying lexical elements. Standard dictionary rules concatenate these readings in multi-branch rules and assign to the whole structure the syntactic and semo-syntactic features described for the general verb system in [1] and under 1. above. Thus, the German standard dictionary rule C 5011 (marked on the following page of computer print-out) analyzes the lexical collocation *ins Gewicht fallen* (be *important*):

C 5011

V
FS(N)
TS(E)
IS(S)
FO(LA)
TO(LA)
IO(LA)

Standard String:	C 4201	C 9028	C 149	C 20868
	fall	*in*	*s*	*Gewicht*

To this standard rule (C 5011) the following German normal form rule applies:

V BE+IMPOR TANT A CAT(V+P) N TM(INS+G EWICHT+F FALLEN)	C 5011	C 20868	C 149 A 1 B 2	C 9028 A 1 B 3	C 4201 A 1 B 4

```
C 5004    V V          C 4403       C 10801
          + TS(HU)     S PX(DAR)
          + IS(S)
          + FO(A)
          + TO(E)
          + IO(U)
          + FS(N)

C 5009    V V          C 4187       C 164        C 21619
          + TS(HU)
          + IS(S)
          + FO(AUFI)
          + TO(IN,AB
            ,NT)
          + IO(U)
          + FS(N)

C 5010    V V          C 4201       C 9010       C 23170
          + TS(E)      S PX(LA)
          + IS(S)
          + FO(LA)
          + TO LA)
          + FS(N)
          + IO(LA)

C 5011    V V          C 4201       C 9028       C 149        C 20868
          + TS(E)      S PX(LA)
          + IS(S)
          + FO(LA)
          + TO(LA)
          + FS(N)
          + IO(LA)

C 5012    V V          C 4206       C 22551
          + TS(HU)     S PX(LA)
          + IS(S)
          + FO(A)
          + TO(E)
          + IO(U)
          + P(LA)
          + FS(N)

C 5013    V V          C 4206       C 20099
          + TS(HU)     S PX(LA)
          + IS(S)
          + FO(A)
          + TO(IN,AB
            ,NT)
          + IO(U)
          + P(LA)
          + FS(N)
```

The corresponding English normal form rule is in the same translation equivalence class, BE+IMPORTANT:

V BE+IMPORTANT A CAT(V+P)	C 5123	C 20	C 10377 A 1 B 3

The right-hand side of this rule refers to the English standard dictionary rule C 5123, which, in turn, generates the English standard string:

C 5123

V
FS(N)
TS(E)
IS(S)
FO(LA)
TO(LA)
IO(LA)

C 20	C 10377
BE	IMPORTANT

The correct endings and morphological variants (in this case *am*, *are*, *is*, etc.) are generated by the English rearrangement grammar.

German verb phrases which we call "hidden passive phrases"— i.e., those which contain empty function verbs such as *gelangen zu* and *kommen zu*, followed by nominalized verbs— were also treated as lexical collocations. Examples are:

zur Ausstossung gelangen = *be ejected*

zum Einsatz kommen = *be employed*

An additional subscript P identifies these German phrases as passive in meaning to guarantee their correct translation. The English translation equivalents were not coded as phrases.

3.4.2.2 Lexical Collocations with Internal Variables

Some lexical collocations contain variable internal slots, as for example the noun modifier slot in *to take V care that...*, where V stands for "variable": *he took care that...*, *he took* *great* *care that...*, *he took* *the greatest possible* *care that...*, etc. For such phrases, standard rules were written which provide for variables in their right-hand sides (cf. rule C 5120 below). Since the present rule format does not allow optional rule constituents, several rules were coded for such phrases, one for each possible string. As an example, the English standard rules for *take care that* are shown here:

```
C 5119      V V           C 4726         C 23195
            + FS(N)       $ PX(LA)
            + TS(HU)
            + IS(S)
            + FO(CL)
            + TO(TH)
            + IO(O)

C 5120      V V           C 4726      V ADJ      C 23195
            + FS(N)       $ PX(LA)
            + TS(HU)
            + IS(S)
            + FO(CL)
            + TO(TH)
            + IO(O)
```

where C 4726 is the rule number for *take* and its allomorphs and C 23195 the number for *care*.

For some phrases, up to six rules were necessary to allow for optional determiners, noun modifiers, and plural noun endings, e.g., *to pose* (DET)(ADJ) *problem(s)*. These lexical collocations do not constitute set phrases, but rather, instances in which a verb has a specific and unusual meaning (and translation) in the environment of a noun phrase whose head noun is a particular lexical item. Beyond this, all normal rules of NP analysis and generation apply. The development of a new algorithm (lexical collocation phase) is planned to permit the expression of these relations in a more economical manner than the one described above.

For the text *Raketenantriebe: Ihre Entwicklung, Anwendung und Zukunft*, the number of lexical colloction rules coded is approximately as follows:

German standard dictionary rules:	165
English standard dictionary rules:	160
English normal form rules:	140
German normal form rules:	200

3.4.3 Noun Entries

In order to classify German nouns and their translation equivalents adequately, several modifications of the subscripts and their values as described in an earlier report became necessary (cf. [1], pp. II-19 through II-22).

GD = gender (for German only), with the values M (masculine), F (feminine), N (neuter). The subscript SX (sex) is used for English nouns.

FC = form of complement (syntactic), and

TC = type of complement (semantic), replace the earlier OB and TO, respectively.

In addition, the following features were coded:

CL = "paradigmatic class" (1-64 in German, A in English),

ON = "onset" (English only), with the values C (consonant), V (vowel),

IO = "interpretation of object", with the possible values O, O2, and LA (cf. p. III-38, d.),

TT = "tantum noun", with the values S for singular, P for plural (e.g., *Fachleute*, *Kosten*),

CP = "capitalization", with the value N (none), to mark such non-capitalized nouns in German as *sek.*, the abbreviation for *Sekunde*,

LC = "lexical collocation", with the values:

N	noun only
NP	DET and/or ADJ + NO
PN	PREP + NO
PP	PREP + DET (+ADJ) + NO

The following rules are examples of the dictionary rules for one German noun and its two translation equivalents, and the necessary normal form rules for their dictionary items.

German dictionary rule

```
C 20078   VN            * ANREICHE
          + CL(20)        RUNG
          + DG(F)
          + TY(AB)
          + FC(LA,G,
            VON.MIT)
          + TC(LA,IN
            .IN)
          + IO(LA,0.
            02)
          T 1.4
```

English dictionary rule

```
C 21183   V N           * CONCENTR
          + CL(A)         ATION
          + ON(C)     P
          + TY(CN,IN
            )
          + FC(LA'OF
            'OF.WITH
            )/
          + TC(LA'IN
            'IN.IN)/
          + IO(LA'0'
            0.02)
          T 1.4

C 20362   V N           *ENRICHME
          + CL(A)        NT
          + ON(V)        P
          + TY(CN)
```

German normal form rule

```
C 3643   V CONCENTR C 20078
           ATION    $ 10
         $ 2.1
         A CAT(N)
         N TM(ANREI
           CHERUNG)

C 3904   V ENRICHME C 20078
           NT
         A CAT(N)
         N TM(ANREI
           CHERUNG)
```

English normal form rule

```
C 1852   V CONCENTR C 21183
           ATION    $ 10
         $ 2.1
         A CAT(N)

C 3903   V ENRICHME C 20362
           NT
         A CAT(N)
```

Note that in the German and English dictionary rules above, the objective genitive of nouns derived from transitive verbs is indicated under the subscripts FC and TC.

3.4.4 Adjective Entries

The following modifications of the adjective feature system were introduced:

Additions:

CL = "paradigmatic class" (1-20 in German, A for English)

ON = "onset" (English only)

LC = "lexical collocation", with the value A (for German only).

Changes:

FO = "form of object" replaces the earlier OB

TM = "type of modificand" (semantic) (for values, cf. TS under verbs except for IT), and

FM = "form of modificand" (syntactic), with values NO for nominal, CL for clause; these last two subscripts replace the earlier subscript MD

SP = "special adjective", with the possible values PAPL and PRPL (past and present participle, respectively), replaces the earlier subscript FM.

Sample Rules for Adjectives

German dictionary rule

```
C 10807  V A          * GEBUNDEN
         + CL(11)
         + TM(MA'AB
           ,IN+MS)/
         + FO(ANI'L
           A)/
         + TO(IN'LA
           )/
         + IO(O'LA)
         + SP(PAPL)
         T 1.3
```

English dictionary rule

```
C 10807  V A          * COMBINED
         + CL(A)        P
         + TM(AB,IN
           )
         + ON(C)
         ¬

C 10405  V A          * DEPENDEN
         + CL(A)        T
         + TM(AB,PO     P
           )
         + FO(LA,ON
           )
         + TO(LA,AB
           ,PO)
         + IO(LA,O)
         + ON(C)
         T 1.3
```

German transfer rule

```
C 4745   V COMBINED   C 10807
         A CAT(A)
         N TM(GEBUN
           DEN)

C 4744   V DEPENDEN   C 10807
           T          $ 10
         $ 2.1
         A CAT(A)
         N TM(GEBUN
           DEN)
```

English transfer rule

```
C 1437   V COMBINED C 10807
         A CAT(A)

C 1355   V DEPENDEN C 10405
           T        $ 10
         $ 2.1
         A CAT(A)
```

The cardinal numbers from 0 to 9 were coded with the features

TY = "type of number", with the values DG for "digit", SP for "spelled out"

ON = "onset" (English only)

```
V NU          * SIEBEN
+ TY(SP)

V NU          * SIEBEN
+ TY(DG)
```

3.4.5 Revised Statistics of the Lexical Data Base for *Raketenantriebe: Ihre Entwicklung, Anwendung und Zukunft*

German dictionary:
- 900 verb stems
- 165 lexical verb phrases (lexical collocations)
- 860 adjective stems
- 3,180 noun stems

English dictionary:
- 950 verb stems
- 160 lexical verb phrases
- 850 adjective stems
- 3,200 noun stems

German normal form:
- 1,000 verbs
- 200 lexical verb phrases
- 860 adjectives
- 3,180 nouns

English normal form:
- 1,000 verbs
- 140 lexical verb phrases
- 850 adjectives
- 3,200 nouns

In addition to the dictionaries for the text *Raketenantriebe: Ihre Entwicklung, Anwendung und Zukunft*, the compilation of major lexical lists continued. The English-German adjective list now contains approximately 27,300 English adjectives coded with German translation equivalents and subject area or stylistic descriptors. Of these, some 12,000 have been given syntactic and semo-syntactic features.

CONCLUSION

The results of the efforts in systems construction and linguistics performed under contract F30602-73-C-0192 strengthen confidence in the soundness of the theoretical basis of LRS and support the expectation expressed in the feasibility study that quality machine translation can be obtained.

Future efforts will concentrate on the completion of LRS and its application to an operational environment.

R E F E R E N C E S

1. Lehmann, Winfred P., and Rolf Stachowitz. *Development of German-English Machine Translation System.* Austin: Linguistics Research Center, The University of Texas at Austin, March 1972. Report RADC-TR-72-47. (First Annual Report.)

2. ________. *Development of German-English Machine Translation System.* Austin: Linguistics Research Center, The University of Texas at Austin, April 1972. Report RADC-TR-72-74. (Second Annual Report.)

3. ________. *Development of German-English Machine Translation System.* Austin: Linguistics Research Center, The University of Texas at Austin, [July] 1973. (Final Report.)

4. ________. *Research in German-English Machine Translation on Syntactic Level,* Vol. 2. Austin: Linguistics Research Center, The University of Texas at Austin, August 1970. Report RADC-TR-69-368. (Final Report.)

5. Stemmer, Josef. *Raketenantriebe— Ihre Entwicklung, Anwendung und Zukunft.* Zürich: Schweizer Druck- und Verlagshaus AG, 1952.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: RADC-TR-73-309	2. GOVT ACCESSION NO:
3. RECIPIENT'S CATALOG NUMBER:	
4. TITLE (and Subtitle): DEVELOPMENT OF GERMAN-ENGLISH MACHINE TRANSLATION SYSTEM	5. TYPE OF REPORT & PERIOD COVERED: Final Technical Report 5 Feb - 4 Jul 73
	6. PERFORMING ORG. REPORT NUMBER: None
7. AUTHOR(s): Dr. Winfred P. Lehmann; Dr. Rolf Stachowitz	8. CONTRACT OR GRANT NUMBER(s): F30602-73-C-0192
9. PERFORMING ORGANIZATION NAME AND ADDRESS: The University of Texas at Austin, Box 7247, University Station, Austin, TX 78712	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 31025F IDHS 04 14
11. CONTROLLING OFFICE NAME AND ADDRESS: Rome Air Development Center (IRDT), Griffiss Air Force Base, New York 13440	12. REPORT DATE: October 1973
	13. NUMBER OF PAGES: 59
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Same	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION DOWNGRADING SCHEDULE: N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

RADC Project Engineer: Zbigniew L. Pankowicz (IRDT)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Computational Linguistics
German-English Machine Translation
Structural/Descriptive Linguistics
Lexicography
Computer Programming

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The report documents performance on a five month R&D effort oriented toward completion of a pilot system for machine translation of German scientific and technical literature into English. Work on grammer formalism concentrated mainly on increasing the power of the subscript grammer to prevent intermediate "forced" readings. Work in system construction concentrated on the completion of the grammer maintenance programs and on the core of the system programs used by all analysis and production algorithms. The linguistic work

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Abstract Cont'd

concentrated on the coverage of the German surface syntax, the "choice rules" for the generation of the corresponding standard structures (deep structures), and their grammatical description.