Limited Distribution

TR-I-0289

The German Grammar for ASURA

Herbert S. Tropf

December, 1992

ATR Interpreting Telephony Research Laboratories, Kyoto
and
Siemens AG, München

Name: Herbert S. Tropf

Stay at ATR: 4 February 1992 to 16 December 1992

Status: Exchange researcher

Company: Siemens AG
ZFE ST SN 74
Otto–Hahn–Ring 6
D–8000 München 83
Germany

The work described in this report has been obtained through the collaborative research of ATR, Kyoto, and Siemens AG, Munich. This documentation and the grammar described in it belong to both ATR and Siemens AG.

Abstract

In this report the German grammar for the ASURA system is described. It is mainly a documentation for maintenance and extension purposes. After an introduction a sketch of the grammar and its input from transfer is given. A few remarks are made about the generation engine. Then the main linguistic as well as technical aspects of the grammar are described. The last sections concern postprocesses, tools and performance. In the appendices attribute–value paths of the syntactic part of the rules and all generated sentences of the conference registration corpus are listed.

Acknowledgements

I want to express my gratitude to Dr. Akira Kurematsu, president of ATR Interpreting Telephony Research Laboratories, who made my stay at ATR possible, to Tsuyoshi Morimoto, head of the Knowledge and Database Department, who never lost his sense of reality and feasibility in realizing the Japanese component of the C–STAR project in such a short period of time, to Noriyoshi Uratani, who saved me a lot of time by removing all kinds of practical obstacles, above all to Gen–ichiro Kikui, with whom I had a very intense and very fruitful cooperation, to Masami Suzuki and Mark Seligman, who had to endure many requirements from my side with respect to the transfer component. The most special thanks, however, go to Mutsuko Tomokiyo, Yves Lepage, and Christian Boitet, with whom I was able to spend such a nice time at ATR and in Japan—and not only talk about speech translation.

Contents

1 Introduction

This is the documentation of the German grammar for the ASURA system at ATR. The grammar is one of several contributions by Siemens within the C–STAR project. This project is based on cooperation between ATR, CMU, Siemens and University of Karlsruhe. Its aim is a demonstration system for trans–oceanic telephone communication between three sites—each site using a different language: Japanese at ATR in Kyoto, English at CMU in Pittsburgh, and German at Siemens in Munich. The main components of such a system are speech recognition, linguistic analysis, transfer, generation, and speech synthesis. The domain of the actual conversations is conference registration. For this, 12 common example dialogs were composed for each language. The official demonstration of this system will be on January 28th, 1993.

The aim of this documentation is mainly to provide information for facilitating the extension and maintenance of the current grammar. Therefore, rather few theoretical reasonings and justifications will be found. Also, there are only few remarks about the component that provides the input for the grammar and about the component to which the output of the grammar is given (cf. sections 3 and 6). The generation engine is also described very briefly (cf. section 4).

The design and implementation of the grammar was started in February 1992. All 12 dialogs were completely covered at the end of October 1992. At the same time, the environment for grammar writing was developed. Because of changes in the analysis and transfer component a few small changes in the grammar had to be added later on. Here the resulting first official version of the grammar is described. [1]

2 Outline of the Grammar

The grammar is a pure unification grammar in the sense that it does not contain an explicit context–free backbone on categories, like e. g. PATR–II. The treatment of linguistic phenomena is inspired by the work of Shieber [9], Gazdar [2], and mainly Pollard & Sag [7] [8]. Lexical information and its projection plays a crucial role. It might be called an HPSG–like grammar although it differs from that in one major aspect. In the grammar described here, as in ordinary context free grammars, a rule expresses two aspects at the same time: the immediate dominance between mother and daughter nodes and the linear precedence between sister nodes. In HPSG (and GPSG) these two aspects are treated separately.

[1] The grammar was designed and written by the author of this documentation. G. Kikui gave much support in utilizing his generation engine. He also developed the environment for grammar writing. In addition, he gave some valuable advice related to HPSG items as well as to a previous version of this report.

The grammar was primarily written for the generation component of the ASURA system, but is in principle reversible for analysis. Only a few minor items are optimized, particularly for generation purposes (e. g. using empty strings for not fully specified addresses and names or for verb arguments with an empty value in the semantic representation).

The current demo version of ASURA is based on single sentence processing. A component for treating the dialog history, i. e. a specific kind of text understanding is not yet been integrated. Although anaphora and ellipses cannot be resolved properly in analysis or transfer, some defaulting techniques are used in transfer and generation which can provide some kind of anaphora (cf. section 5.5). Another consequence of lacking dialog handling is that topic/focus information cannot be provided.

Conceptually, the grammar is clearly partitioned into syntax and lexicon, although formally lexical entries are only a specific case of normal syntax rules. The lexicon is not a full–form lexicon but a lexicon with "baseforms" which are enriched mainly by morphological information. There is no explicit morphological component in the grammar. Morphology is part of the syntax.

A major aspect with which each German grammar has to deal is word order, which is rather free in German. The skeleton of the word order in German sentences is mainly determined by the position of (the parts of) the verb(s). Here are the basic possibilities:

1. Finite verb in the first position and nonfinite verb (when present) or verb prefix (when present) in last position of the sentence

   This order occurs in yes/no–questions, imperatives and in declarative main clauses preceded by a subclause:

   WIRD *es noch Sashimi* GEBEN*?*
   ESSEN *sie alles* AUF*?*
   GIB *ihnen noch Sashimi!*
   Obwohl sie Sake getrunken hatten, KONNTEN *sie noch Sashimi* ESSEN.

2. Finite verb in the last position immediately preceded by nonfinite verb (when present)

   This order occurs in subclauses only:

   Obwohl sie Sake TRINKEN*, können sie noch Sashimi essen.*
   Obwohl sie Sake GETRUNKEN HATTEN*, konnten sie noch Sashimi essen.*

3. Finite verb in the second position and nonfinite verb (when present) or verb prefix (when present) in the last position

   This order occurs in declarative sentences and in wh–questions:

   Dort GIBT *es noch Sashimi.*
   Dort WIRD *es noch Sashimi* GEBEN.
   Sie ESSEN *alles* AUF.

Wer TRANK *den Sake?*
Wer WILL *noch Sake* TRINKEN*?*
Wer ISST *alles* AUF*?*

In principle, the arguments of a verb (or subcategories) as well as "free" arguments can take any remaining position in the sentence. Only in the case of pronouns is the word order more restricted. In addition, focused single arguments can also be extraposed on the left– or right–hand side of the above given basic sentence structures. Of course, there is some tendency of "unmarked" word ordering, but as soon as focus is involved a rather free word order is possible. [2]

The German grammar for ASURA takes a "normalized" word order as the starting point. It is characterized by verb–last position as that in subclauses. Similar suppositions are made e. g. in Block & Schachtl [1] and Tropf [11]. The nonfinal positions of the finite verb and the position of an argument in the *vorfeld* is obtained by utilizing the slash concept of HPSG and GPSG. The position of the verb arguments are determined by the subcategorization feature in the lexical entries. The position of free arguments is slightly restricted within the syntax.

3 Input from the Transfer Component

The input from the transfer component is one feature structure per sentence with two top level features: `sem` and `prag`. In generating a sentence the feature structure of this `sem` feature is subsumed and unified with the semantic part of the (successfully applied) syntactic rules. The `prag` feature need not necessarily be unified with syntactic rules. However, the `prag` feature information can be unified selectively (cf. subsection 5.2).

Basically `sem` contains

- information about the speech act or *illocutionary force type* of the sentence to be generated, e. g. `questionref`, `questionif`, `request`, `promise` or `inform`;
- a matching of `*speaker*` and `*hearer*` with agent (`agen`) and recipient (`recp`);
- the proposition consisting of
  - an indication of what (kind of) verb is to be generated;
  - the arguments of the verb including their modifications;
  - features for tense, aspect, etc.

[2] For a better insight in this matter a lot of research is still to be done, particularly corpus based investigations that can provide statistical data.

The prag feature of the transfer representation contains:

- information about matching of `*speaker*` and `*hearer*` with agent (agen) and recipient (recp), i. e. a repetition of a part of sem;
- an indication intimicy, used for the difference between *Sie* and *du*;
- an indication for politeness, used e. g. for adding *bitte* or realizing the subjunctive instead of the default indicative in some cases.

For more detailled information concerning the transfer component cf. Hasegawa [3] and Suzuki et al. [10]. Here a simple example of the semantic representation of the transfer component is shown. This feature structure is generated into:

Haben Sie schon ein Anmeldeformular?

```
(TRANSOUT :SURFACE
          "NO.006"
          :STRUCTURE_STRING
          "
;;;=========== Transfer Result ============
[[SEM [[RELN QUESTIONIF]
       [AGEN !X4[[LABEL *SPEAKER*]]]
       [RECP !X1[[LABEL *HEARER*]]]
       [OBJE [[RELN HABEN-V]
              [TENSE PRES]
              [AGEN !X1]
              [OBJE [[PARM !X2[]]
                     [RESTR [[RELN ANMELDEFORMULAR-N]
                             [ENTITY !X2]]]
                     [INDEX [[DETERM INDEFART]
                             [NUMBER SING]
                             [OWNER [[LABEL *UNKNOWN*]]]]]]]
              [TLOC-1 [[PARM !X3[]]
                       [RESTR [[RELN SCHON-ADV]
                               [ENTITY !X3]]]]]]]]]
 [PRAG [[HEARER !X1]
        [INTIMACY LOW]
        [POLITENESS [[DEGREE 3]]]
        [SPEAKER !X4]]]]
")
```

As can be seen in this example, a notorious problem of translating Japanese into e. g. German, namely the use of appropriate determiners (definite or indefinite article, possessive pronoun) in the target language, is essentially solved within the transfer component. To get this information, knowledge about domain dependent restrictions is extensively used.

4 A Few Remarks about the Generation Engine

The activation of rules is determined by the `sem` feature of the syntactic rules (for a detailled description cf. Kikui [4] [5]). The bottom–up or the top–down mode can be chosen.

Each feature name (not necessarily the corresponding value) of the top level of the propagated current semantic feature structure, which originated in the transfer component, must be subsumed and unified by the `sem` feature of the left–hand side of the syntactic rules exactly one time.

When only a *value* of the current feature structure, i. e. not the complete feature structure itself, is propagated, the feature structure shrinks and a new top level must be subsumed and unified. The generation ends when there is no active rule.

There is no restriction on the recursion of rules.

An extension of unification are the operators `:set`, `:not` and `:null`. `:set` as in `[case (:set nom dat acc)]` can be seen as a specific kind of disjunction: One of the values must be unifiable. Similarly, `:not` as in `[type (:not refl special)]` can be seen as a negation: None of the values may be unifiable. The values of these operators must be atomic, i. e. they may not be feature structures. With `:null` as in `[agen :null]` can be checked whether a feature name is actually present. If it is not present, this structure cannot be unified. A "normal" unification would be possible.

5 Grammar

5.1 Main Aspects

5.1.1 Subcategorization

One of the major sources of syntactic information is contained in the `subcat` feature of a lexical entry. As mentioned in section 2 the order of arguments in German is rather free. There are some general tendencies, e. g. indirect object before direct object, but the most important factor is probably the topic/focus relation. Since within the current ASURA system neither topic/focus information nor a dialog history is available for generation the

subcat feature is utilized not only to determine the *kind* of verb arguments but also their *position* within a sentence. There are two main reasons for this latter property: First, because of a better efficiency not all possible word orders should be generated, and second, the corpus to be covered contains, of course, a certain topic/focus structure which has no representation in the current system. So, the most transparent solution is to put this word order information into the lexicon.

The value of a subcat should be seen as a list, i. e. the order of the items is relevant. The list structure contains a head (first) and a corresponding tail (rest) which contains again a head and a tail, etc. The subcat feature of a verb with three arguments looks like following example for the verb *sich wenden an*:

```
[subcat [[first [[syn [[cat pp] [pphead an-1] [agr [[case acc]]]]]
                 [sem ?recp]]]
         [rest [[first [[syn [[cat np] [agr [[case acc]
                                            [number ?number]
                                            [person ?person]]]
                             [type refl]]]
                       [sem [[label ?dummy]]]]]
                [rest [[first [[syn [[cat np]
                                    [agr [[case nom]
                                          [number ?number]
                                          [person ?person]]]
                                    [type (:not special)]]]
                              [sem ?agen]]]
                      [rest end]]]]]]]]]
```

Within the grammar all the arguments of a subcat feature are "consumed" by recursively attaching them to the VP and S (cf. subsection 5.1.3).

The difference between obligatory and optional arguments of a certain verb is realized by having more than only one entry for the verb in question. The difference then only concerns the subcat and sem features.

5.1.2 Free Arguments

Free arguments, i. e. arguments that are neither obligatory nor optional arguments of a verb, in general expressions concerning time, location or manner, are (with some exceptions) not part of the subcat feature. Instead, they are treated within the syntax. Here they can be attached to VP (or S) at nearly any position. As a result, possibly several sentences are generated. However, two kinds of restrictions are made. First, they cannot be attached to a VP when a personal pronoun or reflexive pronoun follows immediately. Thus sentences like:

Hans schreibt schon mir.

are suppressed, although the structure is wellformed if the pronoun is stressed contrastively. [3]

And second, there are some cases in the corpus of the 12 dialogs in which an ADVP consisting of a single word would also be generated at the end of a sentence, like:

Haben Sie das Anmeldeformular schon?

This structure is syntactically wellformed even in written language but sounds somewhat strange if the main stress of the sentence is not on *haben*. Because there is no topic/focus component, in which such information could be utilized, the lexicon is "tuned" in some cases to avoid strange effects in the speech synthesis component. In those cases, the free arguments are changed into optional arguments whose position can then be determined by the subcat feature.

5.1.3 Sentence Structure

The illocutionary force types of the transfer representation are syntactically realized as declarative sentence, imperative sentence, wh–question, yes/no–question, or idiomatic expression. (Idiomatic expressions as sentences,like e. g. *Guten Tag!* are very shallow structures.) What follows is a sketch of the projection along the bar–levels from preterminal V to the starting symbol S2.

- V expands to V1 if (among others) the following basic restrictions are kept (cf. rule `V1-V`):
  - `[type nonseppref]`: does not have a separable prefix;
  - `[vform (:not zuinfinitive)]`: is not a *zu*–infinitve, but instead either `finitive`, `infinitive` or `pastpart` (past participle);
  - `[lexid (:not sein-1 heissen-1)]`: is not a copula.
- VPREF (verb prefix) and a following V form V1 if (among others) it is a separable verb. For efficiency reasons a verb prefix is always separated, even if it is orthographically not, like in *abgeschrieben* (cf. rule `V1-VPREF-V`). A postprocess restores the spelling (cf. section 6).
- The particle *zu* and a following V with `[vform zuinfinitive]` form V1 (cf. rule `V1-PARTICLE-V`).
- A V as past participle followed by the past participle of the AUX *werden* form V1. This is the progressive passive form (cf. rule `V1-V-AUX`).

[3] The relevant restriction is `[lasttype (:not pers refl)]` which is used in rule `VP-XP_free-VP`.

- Finally, a nominative NP or an AP followed by a copula form V1 (cf. rule `V1-XP-V`). In this case nominative NP and AP are subject complements.

V1 can expand to VP directly without specific restrictions (cf. rule `VP-V1`) or it is preceded by an ADVP which is a negation construction, like e. g. *überhaupt nicht* (cf. rule `VP-ADVP_neg-V1`).

Within the VP–S–chain the verb arguments of the `subcat` feature and/or free arguments (cf. subsections 5.1.1 and 5.1.2) are successively attached to VP and S. The possible syntactic categories of these arguments are NP, PP, ADVP, VP or S1. The following sentence as an example of an S1 subcategory is to be found in the 12 dialogs:

Ich erwarte, dass einige der Vortragenden teilnehmen.

In this sentence two `subcat` features must be fully covered (cf. rule `VP-VP-S1`).

An example for a VP subcategory is the sentence:

Wieviel kostet es, mit einem Taxi vom Bahnhof Kyoto zum Konferenzzentrum zu fahren?

In cases like these, in which a V is subcategorizing for a VP, the subject of the VP must be omitted (cf. rule `VP-VP-VP`), i. e. not all arguments of its `subcat` feature are syntactically realized, which normally is a precondition for wellformedness.

The following grammar rules form the VP-chain: `VP-V1`, `VP-ADVP_neg-V1`, `VP-NP_special-VP`, `VP-NP_nom-VP`, `VP-XP-VP`, `VP-VP-VP` and `VP-VP-S1`.

VP and a corresponding argument form S if this argument is the last one of the `subcat` list to be consumed. (There might still be a free argument to be consumed.) The corresponding rules are `S-NP_nom-VP`, `S-XP-VP`, and `S-XP_free-S`.

S together with an argument expands to S if it is the last free argument to be consumed and all arguments of the subcat list are already consumed. (For efficiency, only wh–arguments are admitted for this position.) The corresponding rule is `S-XP_free-S`.

This category S, embracing V and its corresponding arguments as well as free arguments, is attached to AUX (auxiliaries for tense and voice, and modals) on the same bar level. (The names for the relevant rules all begin with `S-S-AUX_....`) Furthermore, the negation of the verb is introduced on this level (cf. rules `S-S_neg...`).

The agreement in person and number between subject and finite verb is handled here on the VP and S level, respectively (cf. `agr` feature in rules `S-NP_nom-VP` and `VP-NP_nom-VP`).

On bar level S1 two main structures are covered:

1. For main clauses the "verb–first" position of the finite verb is established using the "slash" concept (cf. `[vpos v-1]` in rule `S1-VAUX_moved-S` and subsection 5.1.4).

2. For subclauses with as well as without complementizers (COMP) the "verb–last" position of the finite verb is established, leaving the verb position as it is in S without using the slash concept (cf. [vpos v-last] in the rules S1-S, S1-COMP-S...).

On the top level S2, again two main structures are covered:

1. The positions of the verb and the arguments remain unchanged with respect to the corresponding S1 if the main clause is a yes/no–question, an imperative, or a declarative which is preceded by a subclause. (These structures are handled in the first rules of the grammar.)

2. The position of the argument that was a daughter of the S node is changed by using the slash concept. The new position of this argument is now the *vorfeld*, i. e. at the beginning of the sentence before the finite verb. This structure is realized in declaratives that are not preceded by a subclause, and in wh–questions. (These structures are handled in the remainder of the S2 rules.)

In addition to these basic sentence structures, S2 dominates idiomatic expressions which normally have a very flat structure, like e. g. *Auf Wiederhören!* Furthermore, on this bar level sentence connectors like *also gut!* and/or the politeness particle *bitte* are also attached the left or right margin of the sentence.

In the sem feature of all S2 rules the illocutionary force types are covered as well as the matching of *speaker*/*hearer* with agent (agen) or recipient (recp). In addition, the proposition, i. e. mainly verb and arguments, is bound to the feature obje.

The following tree structure is shown to give an example of the different levels between V and S2. The generated sentence is (after some postprocesses):

Ich möchte mich zur Konferenz anmelden.

```
S2 [#2357 EXPANDED  (S2-XP_MOVED-S1_INFPROM$1)]
|--NP [#2358 EXPANDED  (NP-PRONP$2 PRONP_PERS1)]
|  |--PRONP [#2381 LEXIFIED "PRONP_PERS1"  NIL]
|--S1 [#2359 EXPANDED  (S1-VAUX_MOVED-S S-S-AUX_MOECHTEN_MOVE)]
|  |--AUX [#2364 LEXIFIED "m:ochten"  (AUX_MOECHTEN)]
|  |--S [#2365 EXPANDED  NIL]
|     |--S [#2366 EXPANDED  (S-NP_NOM-VP VP-XP-VP VP-XP-VP VP-V1 V1-VPREF-V
|     |  |                   V_ANMELDEN$1!=SEP_HABEN_PASSY_V=)]
|     |  |--NP [#2368 TRACE "<trace>"  NIL]
|     |  |--VP [#2369 EXPANDED  NIL]
|     |     |--NP [#2370 EXPANDED  (NP-PRONP_REFL PRONP_REFL$1 PRONP_REFL)]
|     |     |  |--PRONP [#2380 LEXIFIED "PRONP_REFL"  NIL]
|     |     |--VP [#2371 EXPANDED  NIL]
```

```
|     |        |--PP [#2372 EXPANDED  (PP-P-N1$1 N1-N$1)]
|     |        |  |--P [#2377 LEXIFIED "zur"  (P_ZUR)]
|     |        |  |--N1 [#2378 EXPANDED  NIL]
|     |        |     |--N [#2379 LEXIFIED "Konferenz"  (N_KONFERENZ)]
|     |        |--VP [#2373 EXPANDED  NIL]
|     |           |--V1 [#2374 EXPANDED  NIL]
|     |              |--VPREF [#2375 LEXIFIED "an"  (VPREF_AN)]
|     |              |--V [#2376 LEXIFIED "melden"  NIL]
|     |--AUX [#2367 TRACE "<trace>"  NIL]
|--SIGN [#2360 LEXIFIED "."  NIL]
```

There are two traces within this tree: for the finite verb (AUX) and the daughter NP of S, respectively. These categories are "slashified" and their position changed according to the word order rules described above and the slash concept: AUX as lefthand daughter of S1, and NP as left–hand daughter of S2. The order of the arguments under S and VP reflects the order of the `subcat` list. (The names of the applied rules are given in parentheses.)

5.1.4 Slash Feature and Control Feature

The aim of the slash feature is to determine the starting place and landing site of a category whose position of has to be changed in the final tree. Within this grammar these are the finite verb of a main clause and in some cases an argument that has to be in the *vorfeld* of a main clause.

In terms of grammar writing the slash mechanism works as follows: Those categories of the grammar rules that represent the starting place, the landing site and the corresponding categories of the rules "between" have the top level feature `slash`. The value of the `slash` feature depends on the slashified category and is specified in the slash schema at the beginning of the grammar. Two types of slash categories are used: the finite verb that is V or AUX (`vaux` in the slash schema), and an argument that is NP, PP or ADVP (`xp` in the slash schema). The value in the slash schema for the finite verb consists of the feature `syn`, `sem` and `subcat`, and the value in the slash schema for an argument consists of the features `syn`, `sem` and `defsem`. The slash feature is propagated from the starting place via the intermediate rules up to the landing site, where it is checked wether the value of the propagated slash feature is unifiable with the restrictions of the category that is the landing site. (The rule for the landing site of the finite verb is `S1-VAUX_moved-S`. For the landing site of an argument the rule names begin with `S2-XP_moved-...`)

In this current version of the grammar, two phenomena that could also be handled by slashification are not covered: The first concerns the changing of a category position to the right, i. e. behind the verb, as in the discontinuous structure of:

Er hat Eßstäbchen gekauft aus Plastik.

where the PP belonging to the NP is dislocated, the second concerns extrapositions to the left or the right margin, as in:

Diese Frau hier, sie hat noch nie Sake getrunken.

The `control` feature, also a top level feature like `slash`, works in a similar way. The representation of the transfer component does not indicate when an argument is to be realized as a reflexive pronoun. Instead, it gives identical values for different arguments, like e. g. in the following representation for:

Ich möchte mich zur Konferenz anmelden.

```
(TRANSOUT :SURFACE
          "NO.005"
          :STRUCTURE_STRING
          "
;;;=========== Transfer Result ============
[[SEM [[RELN INFORM]
       [AGEN !X1[[LABEL *SPEAKER*]]]
       [RECP !X3[[LABEL *HEARER*]]]
       [OBJE [[RELN DESIRE]
              [TENSE PRES]
              [OBJE [[RELN ANMELDEN-V]
                     [AGEN !X1] ;;; <<<<<<<<<<<<<<<<<<<<<
                     [OBJE !X1] ;;; <<<<<<<<<<<<<<<<<<<<<
                     [DEST [[PARM !X2[]]
                            [RESTR [[RELN KONFERENZ-N]
                                    [ENTITY !X2]]]
                            [INDEX [[DETERM DEFART]
                                    [NUMBER SING]
                                    [OWNER [[LABEL *UNKNOWN*]]]]]]]]]]]]]
 [PRAG [[HEARER !X3]
        [INTIMACY LOW]
        [POLITENESS [[DEGREE 2]]]
        [SPEAKER !X1]]]]
")
```

The strategy of generating a reflexive pronoun instead of the full argument is as follows: Those rules that contain the possible categories to which a (reflexive) argument refers propagate the control feature provided with the syn and sem feature of the category in question. [4] (The relevant rules are VP-NP_nom-VP and S-NP_nom-VP.) The `control` feature is propagated within the S and VP rules in both directions. If within an S or a

[4] In the current version of the grammar only the subject can be referred to. Other arguments are very rarely referred to by reflexive pronouns, both in written and in spoken German.

VP rule an NP or a PP occurs that has the same sem feature as the propagated control feature and whose syntactic agr feature is unifiable with the propagated control feature then the control schema (at the beginning of the grammar) inserts [type refl], which induces the generation of a reflexive NP or PP.

5.1.5 Nominal Phrase Structure

There are three bar levels for N: N, N1 and NP. N is a terminal category and nothing is attached to it. A single N or Q (quantifier) can expand to N1 (cf. rules N1-N... and N1-Q). On the N1 level pre– and postmodifiers can be attached recursively. In the current version of the grammar no order for these attachments is provided (e. g. relative clause after PP, PP after genitive NP), since such order is accepted as it is given by the transfer representation. [5] Postmodifiers of N1 are genitive NP (cf. rule N1-N1-NP_gen), PP's (cf. rule N1-N1-PP_...), relative clause (cf. rule N1-N1-S_rel) and an NP, which expresses identity, as in *7.000 Yen als Anmeldegebühr* (cf. rule N1-N1-NP_als). Premodifiers of N1 are AP and NUMP (cf. rules N1-NUMP-N1 and N1-AP-N1). A single N1 can expand to NP (cf. rule NP-N1_...). A premodifier D (determiner) or Q together with N1 form a complete NP (cf. rules NP-D_... and NP-Q_...). Pronouns directly expand to NP, i. e. without an intermediate N1 (cf. rule NP-PRONP...). There are also rules for NP coordinations (cf. rules NP-NP-COORD-NP and NP-NP-NP).

5.2 Grammar Format and Rule Format

At the beginning of the grammar some specifications are made. These concern the identification of the start symbol of the grammar (S2) and the cat, lex, slash and sem feature. In addition, the slash schema and the control schema are defined (cf. also subsection 5.1.4).

Conceptually the grammar is clearly partitioned into syntax and lexicon, although formally lexical entries are only a specific case of normal syntax rules. Although the order of the rules is not significant for the generation there is a rough order criterion for the syntactic part and another for the lexical part to facilitate the location of rules. The following top–down order was adhered to in the syntax: rules for S2, S1, S, VP, V1, NP, N1, PP, NUMP, AP and ADVP. In the lexicon the following order was adhered to: AUX, V, ADV, D, Q, NUM, A, N, PRONP, VPREF, P, COMP, COORD, PARTICLE and IDIOM. Within these lexical categories the entries are sorted alphabetically.

The format of a rule can best be described by giving a concrete example:

```
(g_pd
 :name VP-XP_free-VP
 :using (=free_tloc_A-XP-B= =free_mann_A-XP-B=)
```

[5] Such restrictions, if necessary for future versions, could easily be added.

```
:internal_structure (VP_1 XP VP_2)
:annotation
((VP_1 [[syn [[cat vp] [vform ?vform] [agr ?agr] [perfaux ?perfaux]
              [tense ?tense] [mood ?mood] [passiveform ?passiveform]
              [passivable ?passivable] [snegation ?snegation]
              [negmod ?negmod] [vpos ?vpos] [relagr ?relagr]
              [lasttype ?type]]]
        [subcat ?subcat]
        [slash ?slash]
        [control ?control]])
 (XP   [[syn [[cat (:set pp advp)] [wh no] [type ?type] [relc no]]]
        [defsem ?defsem]
        [control ?control]])
 (VP_2 [[syn [[cat vp] [vform ?vform] [agr ?agr] [perfaux ?perfaux]
              [tense ?tense] [mood ?mood] [passiveform ?passiveform]
              [passivable ?passivable] [snegation ?snegation]
              [negmod ?negmod] [vpos ?vpos] [relagr ?relagr]
              [lasttype (:not pers refl)]]]
        [subcat ?subcat]
        [slash ?slash]
        [control ?control]])
))
```

Each rule begins with g_pd and has several identifiers (starting with a colon) with corresponding values. Besides the identifier `:using`, which will be described below, all the identifiers are obligatory.

The value of `:name`, the name of the rule, can be chosen freely. Here the convention of using the order of the category names of the rule separated by a hyphen was followed. Additional parts of the rule name are separated by underscoring.

The first item in the list of `:internal_structure` is a pointer to the left–hand side of the rule. The other items are pointers to the elements of the right–hand side of the rule. Their order is relevant. These pointers are just symbols and are of no linguistic significance. For ease of reference (a variant of) the category names was adopted. The names of these pointers must be unique within the same rule.

The value of the identifier `:annotation` is the body of the rule. It consists of the left–hand side and the right–hand side which are identified by the pointers of `:internal_structure`. The order of these elements in the body of the rule is irrelevant, but to avoid confusion the order given by `:internal_structure` is recommended.

The value of the identifier `:using` refers to a list of templates that form an "overlay" for this rule. A template must reflect the global structure of the rule for which it is used, i. e. the same number of categories. So, instead of explicitely writing the same information

in several rules, one can use a template that is internally integrated into the rule that is using it. Templates must be specified in the grammar before the rules that are using them. Templates are mainly utilized for lexical entries. One of the templates for the example given above is

```
(g_pd
 :name =free_tloc_A-XP-B=
 :type :template
 :internal_structure (A XP B)
 :annotation
 ((A  [[sem ?sem-vp]])
  (XP [[sem ?sem-xp]])
  (B  [[sem ?sem-vp [[tloc ?sem-xp]]]])
  ))
```

The difference between a normal syntactic rule and a lexical entry is that a lexical entry has no left– and righthand side but only one category, as e. g. in

```
(g_pd
 :name P_zwischen
 :internal_structure P
 :annotation
 ((P [[syn [[cat p] [type pre] [dclitic no] [agr [[case (:set dat acc)]]]
           [lexid zwischen-1] [lex "zwischen"]]]
     [sem [[label *p-dummy*]]]])))
```

The body of a rule consists of *feature structures* which represent the elements of the left– and right–hand site of a rule. A feature structure is a bracketed expression, which has a feature name (or attribute) and a value. The value can be atomic, as e. g. in `[type pre]` or again a feature structure consisting of attribute and value etc., as e. g. in `[agr [[number sing] [person 3] [gender neu]]]`. Top level feature names are `syn` and `sem`. The value of a feature name can be a variable which is explicitely bound, as e. g. in `[agr ?agr [[case nom] [number ?number]]]` or `[type ?type (:not special)]`. If a variable is not explicitely bound, like e. g. in `[vform ?vform]`, and the value is not instantiated by unification, then this variable is bound by default to the empty feature structure `[]`.

5.3 Terminal and Non–Terminal Categories

There are some special features: `syn` and `sem` are obligatory, `slash` and `control` apply a special schemata (cf. subsection 5.1.4). The `defsem` feature inserts a value for a `sem`

feature whose value is the empty feature structure (cf. subsection 5.5). The `lex` feature is checked, to determine whether it has a value other than `[]`. If it has such a value, then the node will be terminal. If in a lexical entry the `lex` feature has an empty value, then this entry must be specified by the top level feature `[terminal +]`. Non–terminal nodes are either propagation nodes or non–propagation nodes. If the `sem` features of the mother node and a daughter node are identical then it is a propagation node, else it is a non–propagation node.

Following are the lists of terminal and non–terminal categories, in fact the *path* to the corresponding values. They are only mentioned and briefly commented to be easier identified by the grammar writer. For the unification algorithm the cat feature has no special status.

Terminal categories:

```
syn - cat - a -        ;;; particle
syn - cat - adv -      ;;; adverb
syn - cat - aux -      ;;; auxiliary/modal
syn - cat - comp -     ;;; complementizer
syn - cat - coord -    ;;; coordinator
syn - cat - d -        ;;; determiner
syn - cat - idiom -    ;;; idiom
syn - cat - n -        ;;; noun
syn - cat - num -      ;;; numeral
syn - cat - p -        ;;; preposition
syn - cat - particle - ;;; particle
syn - cat - pronp -    ;;; pronoun
syn - cat - q -        ;;; quantifier
syn - cat - sign -     ;;; punctuation mark
syn - cat - v -        ;;; verb
syn - cat - vpref -    ;;; verb prefix
```

Non–terminal categories:

```
syn - cat - advp -     ;;; adverbial phrase
syn - cat - ap -       ;;; adjective phrase
syn - cat - n1 -       ;;; noun (bar 1)
syn - cat - np -       ;;; noun phrase
syn - cat - nump -     ;;; numeral phrase
syn - cat - pp -       ;;; prepositional phrase
syn - cat - s -        ;;; sentence (bar 0)
syn - cat - s1 -       ;;; sentence (bar 1)
syn - cat - s2 -       ;;; sentence (bar 2)
syn - cat - v1 -       ;;; verb (bar 1)
syn - cat - vp -       ;;; verb phrase
```

Complete attribute–value paths for top level features are to be found in appendix A.

5.4 Specific Syntactic Phenomena

5.4.1 Subject Complements

Subject complements, which can be either a nominative NP or a non–inflected AP, are directly attached to V, which is a copula (cf. subsection 5.1.3). The result is a V1. Examples of the corpus are e. g.:

Mein Name ist MAYUMI SUZUKI.
Bitte wenden Sie sich jederzeit an mich, wenn etwas UNKLAR *ist!*

5.4.2 Lexical and Non–lexical Reflexive Pronouns

There are two types of reflexive pronouns: lexical and non–lexical. A lexical reflexive pronoun is coded as an element in the subcat list without having a semantic counterpart in the transfer representation. An example is the verb *sich spezialisieren*, where the case of the reflexive pronoun must be accusative and number as well as person must agree with the corresponding features of the subject, which is a nominative NP:

```
(g_pd
 :name V_spezialisieren_refl
 :using (=nsep_haben_passn_V=)
 :internal_structure V
 :annotation
 ((V [[syn [[lexid spezialisieren-1] [lexfull "spezialisieren"]]]
      [sem [[reln spezialisieren-v] [agen ?agen] [obje ?obje]]]
      [subcat [[first [[syn [[cat pp] [pphead auf-1] [agr [[case acc]]]]]
                       [sem ?obje]]]
               [rest [[first [[syn [[cat np] [agr [[case acc]
                                                  [number ?number]
                                                  [person ?person]]]
                                   [type refl]]] ;;;<<<<<<<<<<<<<<<<<<<<
                             [sem [[label ?dummy]]]]]
                      [rest [[first [[syn [[cat np]
                                           [agr [[case nom]
                                                 [number ?number]
                                                 [person ?person]]]
                                           [type (:not special)]]]
                                     [sem ?agen]]]
```

```
                                    [rest end]]]]]]]]]]))
```

The treatment of non–lexical reflexive pronouns is described in subsection 5.1.4 above.

5.4.3 Stative and Progressive Passive

The transfer representation indicates directly when and what type of passive is to be generated. There are two types of passive in German: stative passive (*Zustandspassiv*), like e. g. in

Bitte überweisen Sie auf das Konto, das in der Ankündigung ANGEGEBEN IST*!*

and progressive passive (*Vorgangspassiv*), like e. g. in

Ich schicke Manuskriptformulare mit, wenn der Beitrag AKZEPTIERT WORDEN IST.

Both types occur in the corpus. The "passive transformation" is basically done in the lexicon. For verbs that can have passive voice there is in principle an entry for active voice and another entry for passive voice. Because in the corpus only some verbs are realized in passive voice most of the verbs are only coded for active voice. The difference between the two voices is reflected only by the subcat and sem features. In general, in the active voice the agent is nominative and the object (if present) accusative whereas in the passive voice the object is nominative and the agent unspecified or realized as a specific PP, in German *von* or *durch*. This "transformation" is reflected in the lexicon by using two different entries for a verb, as e. g. in *halten*:

```
(g_pd
 :name V_halten
 :using (=nsep_haben_passy_V=)
 :internal_structure V
 :annotation
 ((V [[syn [[lexid halten-1] [lexfull "halten"]]]
      [sem [[reln halten-v] [agen ?agen] [obje ?obje]]]
      [subcat [[first [[syn [[cat np] [agr [[case acc]]]
                             [type (:not special)]]]
                       [sem ?obje]]]
               [rest [[first [[syn [[cat np] [agr [[case nom]]]
                                    [type (:not special)]]]
                              [sem ?agen]]]
                      [rest end]]]]]]]))

(g_pd
 :name V_halten_passive
 :using (=nsep_haben_passy_V=)
```

```
:internal_structure V
:annotation
((V [[syn [[lexid halten-1] [lexfull "halten"]]]
     [sem [[reln halten-v] [agen :null] [obje ?obje]]]
     [subcat [[first [[syn [[cat np] [agr [[case nom]]]
                           [type (:not special)]]]
                     [sem ?obje]]]
             [rest end]]]]))
```

For both types of passive a rule is applied which attaches S to an AUX (cf. rules beginning with `S-S-AUX_auxil...` and subsection 5.1.3). The only difference is in V1: For the stative passive only V with `[vform pastpart]` is expanded to V1, whereas for the progressive passive V and AUX *werden* (both `[vform pastpart]`) are expanded to V1 (cf. rules `V1-V` and `V1-V-AUX`).

5.4.4 Relative Clause

Like in the case of non–lexical relative pronouns there is no direct indication in the transfer representation that an argument should be realized as a relative pronoun. Instead, there is a pointer to the full representation of the argument which is modified by the relative clause. However, the structure of the transfer representation tells, that it is suited for a relative clause.

A relative clause is dominated by S and attached to the right of N1. In rule `N1-N1-S_rel` it is stated that the relative pronoun and N1, the head of the rule, must agree in number and gender. So it is necessary that the subcat feature of the verb of the relative clause fits this restriction, i. e. the argument which is consumed last (by an `S-...` rule) must match the agreement restriction.

Restrictive and non–restrictive relative clauses are neither differentiated in the transfer representation nor in the syntax.

5.4.5 Other Subclauses and *zu*–Infinitive

Other subclauses, which have a complementizer, are dominated by S1 (cf. rules `S1-COMP-S...` and subsection 5.1.3). In this version of the grammar only subordinating complementizers, that induce a verb–last position for the subclause are covered. Coordinating complementizers (e. g. *denn*), i. e. coordinators that do not induce verb–last position, are not covered.

The handling of the *zu*–infinitive is described within subsection 5.1.3 above.

5.4.6 S–Negation and NP–Negation

The negation of the verb/sentence and the negation of an argument is indicated in the transfer representation in the same way ([RELN NEGATE]). In this representation the negation of the verb/sentence can, in addition, have a modifier on the same level (e. g. *überhaupt nicht*). This information about negation is propagated down from the S level (cf. rules S-S_neg...) to a VP which expands to ADVP, the structure for negation, and V1 (cf. rule VP-ADVP_neg-V1). At this point only V1 (not ADVP) is slashified. This guarantees a correct word order for negated sentences even when the finite verb is not in verb–last position, as e. g. in

Ich verstehe Japanisch überhaupt nicht.
Ich habe Japanisch überhaupt nicht verstanden.

A negated argument is realized using a quantifier (cf. rule NP-Q_kein-N1...), as e. g. in

Es gibt diesmal keinen Rabatt.

No *nicht*–construction is used for NP–negation.

5.4.7 Names, Addresses and Dates

Names, addresses and dates have a normalized structure in the transfer representation. That means each token has the maximal number of feature names for the item in question. However, when e. g. the first name or the year is not specified, feature names may have an empty value. To avoid a lot of rules for all the possible combinations of specified and unspecified feature names there is only one rule for names, addresses and dates, respectively. Those features of the transfer representation that have an empty value are generated as an empty string.

5.4.8 Idioms

The term "idiom" as it is used here means an expression consisting of more than one word which is given as a single symbol by the transfer component. Examples are *Internationales Konferenzzentrum Kyoto; so schnell wie möglich; Kann ich Ihnen helfen? Vielen Dank!* etc.

Many of them, like e.g. *Vielen Dank!* are totally fixed structures and equivalent to a complete sentence. These are directly dominated by S2, i. e. basically remain unanalyzed. Other cases, like e.g. *Internationales Konferenzzentrum Kyoto*, are not sentence equivalents and a decomposition is necessary because some parts can be inflected.

5.4.9 Cliticization

There is no indication in the transfer component when a definite article should be cliticized after certain prepositions, like e. g. *am* vs. *an dem*, or *ins* vs. *in das*. When there is a PP argument in the subcat list whether there should be a cliticization of the definite article (if a cliticization is possible at all) or whether there should be no cliticization can be explicitly specified by the `dclitic` feature . If `dclitic` is not specified, both the cliticized and the non–cliticized version are generated.

5.5 Customization

For the generation of some syntactic structures no indication is given by the transfer output. In these cases a decision has to be made by the generation component. The general strategy followed was: The "normal" structure is default, and for specific structures additional features or rules are used.

Default settings are e. g. [`mood indicative`] for main clauses and subclauses, [`degree pos`], i. e. positive as degree of comparison for adjectives, and [`number sing`] for speaker/hearer. For [`tense past`] as an indication of the transfer representation only perfect tense (no preterite, no past perfect) is generated, which in general corresponds to the common use in spoken German.

Occasionally, in the transfer representation, an argument has an empty value, as in the following example for

Ich möchte Sie etwas fragen.

```
(TRANSOUT :SURFACE
          "NO.185"
          :STRUCTURE_STRING
          "
;;;=========== Transfer Result ============
[[SEM [[RELN INFORM]
       [AGEN !X2[[LABEL *SPEAKER*]]]
       [RECP !X3[[LABEL *HEARER*]]]
       [OBJE [[RELN DESIRE]
              [TENSE PRES]
              [OBJE [[RELN FRAGEN-V]
                     [AGEN !X2]
                     [obje []] ;;;<<<<<<<<<<<<<<<<<<<<<<<<
                     [RECP !X3]]]]]]]
 [PRAG [[HEARER !X3]
        [INTIMACY LOW]
```

```
        [POLITENESS [[DEGREE 2]]]
        [SPEAKER !X2]]]]
")
```

If such an argument corresponds to an optional or a free argument for the German verb then it is not generated at all (although it must be unified, e. g. in the lexical entry). But if the empty argument of the transfer representation is an obligatory argument for the corresponding German verb a "dummy" value is inserted by use of the `defsem` feature. Depending on the other arguments in the `subcat` list these dummy values are the pronouns *das*, *es* and *etwas*.

In the case of names, addresses and dates empty strings are generated when a certain item is missing (cf. subsection 5.4.7).

To improve the quality of the currently used speech synthesis with respect to Japanese and English proper names, the correct spelling has been changed in some cases.

6 Completing the Final Output String

Syntax and lexicon provide the complete morphological information, besides the corresponding output strings for inflectional suffixes and *umlaute*. For efficiency the graph, which is built up by applying the rules of the grammar, contains as terminal nodes the *baseforms*, i. e. the `lex` features together with the complete semantic, syntactic (and morphological) information of these nodes. The *inflected* forms are provided by an additional component using this information of the terminal nodes where necessary. This component is a finite state transducer which has access to the graph.

And finally, there are some postprocesses on the hitherto generated string that concatenate VPREF and V, make empty strings invisible, remove blanks before punctuation marks and reduce multiple successive commas, a result of empty items in names, addresses and dates (cf. subsections 5.4.7 and 5.5).

7 Tools

There are several tools available for debugging:

- The debug level can be changed according to the amount of information that should be made available.
- Most important is the so–called "s_checker" which makes it possible to test specific rules for application on a certain feature structure.

- There are break facilities with which it is possible to inspect all the information on a specific node of the graph.
- Generation results can be compared.
- Specific features can be added to the tree printer.

In addition, there exist some tools for monitoring the results of test runs.

8 Performance

All 262 sentences of the 12 dialogs (and 72 sentences of 3 additional demo dialogs) are covered by the grammar. All generated sentences including variants are syntactically wellformed. In this sense the grammar is reliable and robust.

There are 145 syntactic rules and 420 lexical entries in this first version of the grammar.

For the 262 input structures of the transfer component 279 sentences were generated.

For each of these 262 input structures the run time for generation on a SUN SPARC–2 (32 MB memory) was 1.85 sec, on average.

Other statistical values are:

Total Run Time	484.590	sec
Maximum Run Time	19.780	sec (ID=NO.170)
Total Space Consed	154,551,488	bytes
Maximum Space Consed	4,322,184	bytes (ID=NO.170)
PD Access	18,364	
Cache Fault	0	
PD Activated	25,012	
PD Applied	18,860	
number of PM entries	3,749	
Assumption Node	786	
Sub Tree	5,098	
Output	279	
Input	262	

A Attribute–Value Paths

These are the paths beginning at the top level features. Not included are paths for the top level features subcat and sem as well as paths that indicate lexical strings. [6]

```
control -
control - sem -
control - syn - agr - gender -
control - syn - agr - number -
control - syn - agr - person -
defsem -
defsem - label - *xp-dummy* -
env - intimacy - high -
env - intimacy - low -
env - politeness - degree -
env - politeness - degree - 3 -
slash -
slash - :null -
slash - vaux - sem -
slash - vaux - subcat -
slash - vaux - syn - agr -
slash - vaux - syn - cat -
slash - vaux - syn - vform - finite -
slash - xp -
slash - xp - defsem -
slash - xp - sem -
slash - xp - syn - cat -
syn - addmodal -
syn - addmodal - dummy -
syn - addmodal - koennen-1 -
syn - agr -
syn - agr - case -
syn - agr - case - acc -
syn - agr - case - dat -
syn - agr - case - gen -
syn - agr - case - nom -
syn - agr - gender -
syn - agr - gender - fem -
syn - agr - gender - mas -
syn - agr - gender - neu -
```

[6] The paths are extracted from the internal representation of the grammar. The program is written by Lepage using his C–toolbox (cf. Lepage [6]).

```
syn - agr - number -
syn - agr - number - plur -
syn - agr - number - sing -
syn - agr - person -
syn - agr - person - 1 -
syn - agr - person - 2fam -
syn - agr - person - 2pol -
syn - agr - person - 3 -
syn - agr - strength -
syn - agr - strength - strg -
syn - agr - strength - weak -
syn - auxused - no -
syn - auxused - yes -
syn - comp -
syn - comp - dummy -
syn - dclitic -
syn - dclitic - no -
syn - dclitic - yes -
syn - degree -
syn - degree - pos -
syn - index -
syn - index - determ -
syn - index - determ - indefart -
syn - index - number -
syn - index - number - plur -
syn - index - number - sing -
syn - index - owner -
syn - index - owner - label - *unknown* -
syn - lasttype -
syn - lasttype - dummy -
syn - lasttype - special -
syn - mood -
syn - mood - imperative -
syn - mood - indicative -
syn - mood - subjunctive -
syn - negmod -
syn - negmod - parm -
syn - negmod - restr - entity -
syn - negmod - restr - reln -
syn - negmod - restr - reln - dummy -
syn - partaderived - no -
syn - partaderived - yes -
syn - passivable -
```

```
syn - passivable - no -
syn - passivable - yes -
syn - passiveform -
syn - passiveform - dummy -
syn - passiveform - progpast -
syn - passiveform - progpres -
syn - passiveform - statpres -
syn - perfaux -
syn - perfaux - haben-1 -
syn - perfaux - sein-1 -
syn - pphead -
syn - pphead - auf-1 -
syn - pphead - bei-1 -
syn - pphead - biszu-1 -
syn - pphead - fuer-1 -
syn - pphead - in-1 -
syn - pphead - in_bezug_auf-1 -
syn - pphead - nach-1 -
syn - pphead - pro-1 -
syn - pphead - von-1 -
syn - pphead - vor-1 -
syn - pphead - zu-1 -
syn - pref -
syn - pref - ab-1 -
syn - pref - an-1 -
syn - pref - aus-1 -
syn - pref - ein-1 -
syn - pref - fest-1 -
syn - pref - hinein-1 -
syn - pref - mit-1 -
syn - pref - nach-1 -
syn - pref - rueckgaengig-1 -
syn - pref - teil-1 -
syn - pref - vor-1 -
syn - pref - zurueck-1 -
syn - relagr -
syn - relagr - gender -
syn - relagr - number -
syn - relagr - person - 3 -
syn - relc -
syn - relc - no -
syn - relc - yes -
syn - snegation -
```

```
syn - snegation - no -
syn - snegation - yes -
syn - tense -
syn - tense - dummy -
syn - tense - future -
syn - tense - past -
syn - tense - pres -
syn - type -
syn - type - attr -
syn - type - auxil -
syn - type - card -
syn - type - common -
syn - type - compar -
syn - type - defart -
syn - type - demonstr-near -
syn - type - dummy -
syn - type - indef -
syn - type - indefart -
syn - type - mass -
syn - type - modal -
syn - type - nonseppref -
syn - type - nonwh -
syn - type - notsequiv -
syn - type - ord -
syn - type - pers -
syn - type - poss -
syn - type - pre -
syn - type - proper -
syn - type - quant -
syn - type - refl -
syn - type - rel -
syn - type - seppref -
syn - type - sequiv -
syn - type - special -
syn - type - subord -
syn - type - wh -
syn - vform -
syn - vform - finite -
syn - vform - infinitive -
syn - vform - pastpart -
syn - vform - zuinfinitive -
syn - vpos -
syn - vpos - v-1 -
```

syn - vpos - v-last -
syn - wh -
syn - wh - no -
syn - wh - yes -
terminal - + -

B Generated Sentences

These are the 262 sentences of the 12 dialogs of the conference registration corpus generated with the first official version of the German grammar for ASURA. (Variants are included but not counted.)

```
NO.001 : "Hallo."
NO.002 : "Ist dort das Konferenzsekretariat?"
NO.003 : "Ja."
NO.004 : "Das ist richtig."
NO.005 : "Ich m:ochte mich zur Konferenz anmelden."
NO.006 : "Haben Sie schon ein Anmeldeformular?"
NO.007 : "Nein."
NO.008 : "Noch nicht."
NO.009 : "Ich verstehe."
NO.010 : "Also gut, ich schicke Ihnen ein Anmeldeformular."
NO.011 : "Kann ich Ihre Adresse und Ihren Namen haben, bitte?"
NO.012 : "Meine Adresse ist dreiundzwanzig, Tschaja-matschi,
          Kitta-ku, O:saka."
NO.013 : "Mein Name ist Ma-jumi Suhsuhki."
NO.014 : "Ich verstehe."
NO.015 : "Ich werde Ihnen ein Anmeldeformular sofort schicken."
       : "Ich werde Ihnen sofort ein Anmeldeformular schicken."
NO.016 : "Bitte wenden Sie sich jederzeit an mich, wenn etwas unklar
          ist!"
NO.017 : "Vielen Dank."
NO.018 : "Auf Wiederh:oren."
NO.019 : "Auf Wiederh:oren."
NO.020 : "Hallo."
NO.021 : "Hier ist das Konferenzsekretariat."
NO.022 : "Ich m:ochte an der Konferenz teilnehmen."
NO.023 : "Was soll ich machen?"
NO.024 : "Sie m:ussen zuerst ein Anmeldeformular verwenden."
NO.025 : "Haben Sie schon ein Anmeldeformular?"
NO.026 : "Noch nicht."
NO.027 : "Bitte schicken Sie mir das Formular!"
NO.028 : "Also gut, kann ich Ihre Adresse und Ihren Namen haben,
          bitte?"
NO.029 : "Meine Adresse ist eins, zwei, Tockui-matchi, Higaschi-ku,
          O:saka."
NO.030 : "Mein Name ist Taro Schimisu."
NO.031 : "Ich verstehe."
NO.032 : "Wird eine Teilnahmegeb:uhr verlangt?"
```

NO.033 : "Ja."
NO.034 : "F:unfunddrei:sigtausend Jenn pro Person werden als Anmeldegeb:uhr verlangt."
NO.035 : "Ich verstehe."
NO.036 : "Vielen Dank."
NO.037 : "Auf Wiederh:oren."
NO.038 : "Hallo."
NO.039 : "Ist dort das Konferenzsekretariat?"
NO.040 : "Ja."
NO.041 : "Das ist richtig."
NO.042 : "Kann ich Ihnen helfen?"
NO.043 : "Ich m:ochte mich zur Konferenz anmelden."
NO.044 : "Wie soll ich vorgehen?"
NO.045 : "Bitte verwenden Sie ein Anmeldeformular!"
NO.046 : "Haben Sie schon ein Anmeldeformular?"
NO.047 : "Nein."
NO.048 : "Noch nicht."
NO.049 : "Ich verstehe."
NO.050 : "Also gut, ich schicke Ihnen ein Anmeldeformular."
NO.051 : "Kann ich Ihre Adresse und Ihren Namen haben, bitte?"
NO.052 : "Meine Adresse ist dreiundzwanzig, Tschaja-matschi, Kitta-ku, O:saka."
NO.053 : "Mein Name ist Ma-jumi Suhsuhki."
NO.054 : "Ich verstehe."
NO.055 : "Ich werde Ihnen ein Anmeldeformular sofort schicken."
: "Ich werde Ihnen sofort ein Anmeldeformular schicken."
NO.056 : "Vielen Dank."
NO.057 : "Auf Wiederh:oren."
NO.058 : "Ja."
NO.059 : "Hier ist das Konferenzsekretariat."
NO.060 : "Ich m:ochte, da:s Sie mich :uber die Teilnahmegeb:uhr f:ur die Konferenz informieren."
NO.061 : "Wieviel betr:agt die Teilnahmegeb:uhr, wenn ich mich jetzt zur Konferenz anmelde?"
NO.062 : "Ja."
NO.063 : "Die Teilnahmegeb:uhr betr:agt zur Zeit f:unfunddrei:sigtausend Jenn pro Person."
NO.064 : "Vierzigtausend Jenn betr:agt es, wenn Sie sich n:achsten Monat anmelden."
NO.065 : "Die Tagungsb:ande und die Empfangskosten sind in der Teilnahmegeb:uhr enthalten."
NO.066 : "Ich bin ein Mitglied der Gesellschaft f:ur Informatik."
NO.067 : "Gibt es keinen Rabatt auf die Teilnahmegeb:uhr?"

```
NO.068 : "Es gibt keinen Rabatt diesmal."
       : "Es gibt diesmal keinen Rabatt."
NO.069 : "Ich verstehe."
NO.070 : "Wie soll ich die Teilnahmegeb:uhr bezahlen?"
NO.071 : "Bitte bezahlen Sie die Teilnahmegeb:uhr mit einer
          Bank:uberweisung!"
NO.072 : "Bitte :uberweisen Sie auf das Konto, das in der Ank:undigung
          angegeben ist!"
NO.073 : "Und, Ende des Jahres ist der Einsendeschlu:s."
NO.074 : "Ich verstehe."
NO.075 : "Vielen Dank."
NO.076 : "Gerne geschehen."
NO.077 : "Bitte wenden Sie sich an mich jederzeit, wenn etwas unklar
          ist!"
NO.078 : "Auf Wiederh:oren."
NO.079 : "Ja."
NO.080 : "Hier ist das Konferenzsekretariat."
NO.081 : "Ich m:ochte einen Vortrag auf der Konferenz halten."
NO.082 : "Bitte informieren Sie mich :uber das Thema der Konferenz!"
NO.083 : "Diese Konferenz umfa:st ein weites Forschungsgebiet in Bezug
          auf das Telefondolmetschen."
NO.084 : "Ich erwarte, da:s Leute, die sich auf Linguistik und
          Psychologie spezialisieren, teilnehmen."
NO.085 : "Ich verstehe."
NO.086 : ":Ubrigens, was ist die offizielle Sprache auf der Konferenz?"
NO.087 : "Das ist Englisch und Japanisch."
NO.088 : "Japanisch verstehe ich :uberhaupt nicht."
       : "Ich verstehe Japanisch :uberhaupt nicht."
NO.089 : "Gibt es eine Simultan:ubersetzung ins Englische, wenn etwas
          auf Japanisch vorgetragen wird?"
       : "Gibt es eine Simultan:ubersetzung in das Englische, wenn
          etwas auf Japanisch vorgetragen wird?"
NO.090 : "Ja."
NO.091 : "Ich bereite eine Simultan:ubersetzung ins Englische vor."
       : "Ich bereite eine Simultan:ubersetzung in das Englische vor."
NO.092 : "Ich verstehe."
NO.093 : "Vielen Dank."
NO.094 : "Auf Wiederh:oren."
NO.095 : "Hier ist das Konferenzsekretariat."
NO.096 : "Bitte sagen Sie mir Details :uber die Konferenz!"
NO.097 : "Haben Sie die Ank:undigung f:ur die Konferenz?"
NO.098 : "Nein."
NO.099 : "Ich habe das nicht."
```

NO.100 : "Ich verstehe."
NO.101 : "Die Konferenz wird vom zweiundzwanzigsten August bis zum f:unfundzwanzigsten im internationalen Konferenzzentrum Kyoto abgehalten."
NO.102 : "Die Teilnahmegeb:uhr betr:agt vierzigtausend Jenn."
NO.103 : "Bitte reichen Sie eine Zusammenfassung bis zum zwanzigsten M:arz ein, wenn Sie etwas vortragen m:ochten!"
NO.104 : "Ich schicke Ihnen die Ank:undigung f:ur die Konferenz. Bitte werfen Sie einen Blick hinein!"
NO.105 : "Kann ich Ihren Namen und Ihre Adresse haben, bitte?"
NO.106 : "Das ist Adam Smi:s."
NO.107 : "Meine Adresse ist siebenundzwanzig, sieben, zwei, Tammattsuhkuri, Higaschi-ku, O:saka."
NO.108 : "Ich verstehe."
NO.109 : "Kann ich Ihre Telefonnummer haben, bitte?"
NO.110 : "Ja."
NO.111 : "Das ist drei, sieben, zwei, acht, null, eins, acht."
NO.112 : "Das ist drei, sieben, zwei, acht, null, eins, acht, nicht wahr?"
NO.113 : "Ja."
NO.114 : "Das ist richtig."
NO.115 : "Also gut, vielen Dank."
NO.116 : "Auf Wiederh:oren."
NO.117 : "Ja."
NO.118 : "Hier ist das Konferenzsekretariat."
NO.119 : "Ich habe eine Bitte."
NO.120 : "Ich habe mich zur Konferenz angemeldet."
NO.121 : "Ich m:ochte meine Teilnahme r:uckg:angigmachen."
NO.122 : "K:onnte ich Ihren Namen haben, bitte?"
NO.123 : "Ja."
NO.124 : "Das ist Dschimm Waibel von Bell L:abs."
NO.125 : "Sie haben f:unfundachtzigtausend Jenn als Anmeldegeb:uhr schon :uberwiesen, nicht wahr?"
: "Sie haben schon f:unfundachtzigtausend Jenn als Anmeldegeb:uhr :uberwiesen, nicht wahr?"
: "Sie haben f:unfundachtzigtausend Jenn als Anmeldegeb:uhr schon :uberwiesen, nicht wahr?"
NO.126 : "Ja."
NO.127 : "Das ist richtig."
NO.128 : "K:onnen Sie die Anmeldegeb:uhr zur:uckerstatten, bitte?"
NO.129 : "Es tut mir sehr leid, das ist nicht m:oglich."
NO.130 : "In der Ank:undigung steht."
NO.131 : "Eine Zur:uckzahlung f:ur eine Abmeldung nach dem

siebenundzwanzigsten September ist nicht m:oglich."
NO.132 : "Ich schicke Ihnen das Programm und die Tagungsb:ande sp:ater."
: "Ich schicke Ihnen sp:ater das Programm und die Tagungsb:ande."
NO.133 : "Also gut, kann anstatt meiner jemand anders teilnehmen?"
NO.134 : "Das macht nicht viel aus."
NO.135 : "Bitte informieren Sie mich vorher, wenn ein Stellvertreter teilnimmt!"
NO.136 : "Ich verstehe."
NO.137 : "Ich informiere Sie, wenn ein Stellvertreter feststeht."
NO.138 : "Auf Wiederh:oren."
NO.139 : "Ja."
NO.140 : "Hier ist das Konferenzsekretariat."
NO.141 : "Ich habe geh:ort, da:s es eine Stadtrundfahrt w:ahrend der Konferenz gibt."
NO.142 : "Kann ich noch teilnehmen?"
: "Kann ich noch teilnehmen?"
NO.143 : "Ja."
NO.144 : "Sie k:onnen noch teilnehmen."
: "Sie k:onnen noch teilnehmen."
NO.145 : "Am f:unften August nachmittags besichtige ich den Ki-joh-mie-su-dera-Tempel, den Kinnkahkuh-dschi-Tempel und den Ryoh-ahn-dschi-Tempel."
NO.146 : "Nehmen Sie teil?"
NO.147 : "Wieviel betr:agt die Teilnahmegeb:uhr?"
NO.148 : "Achttausend Jenn betr:agt es."
NO.149 : "Das Abendessen ist in der Teilnahmegeb:uhr enthalten."
NO.150 : "Nehmen die Vortragenden teil?"
NO.151 : "Ich erwarte, da:s einige der Vortragenden teilnehmen."
NO.152 : "Ich verstehe."
NO.153 : "Also gut, ich m:ochte teilnehmen."
NO.154 : "Also gut, kann ich Ihren Namen und die Anzahl der Personen in Ihrer Gruppe haben, bitte?"
NO.155 : "Das ist Ma-jumi Suhsuhki."
NO.156 : "Ich nehme mit meiner Frau teil."
NO.157 : "Der Treffpunkt ist vor der Rezeption des Konferenzzentrums."
NO.158 : "Bitte bezahlen Sie die Teilnahmegeb:uhr dann am Treffpunkt!"
NO.159 : "Ich verstehe."
NO.160 : "Vielen Dank."
NO.161 : "Also gut, ich erwarte Sie."
NO.162 : "Ja."
NO.163 : "Hier ist das Konferenzsekretariat."

NO.164 : "Ich m:ochte Sie etwas :uber die Themen, die in der Konferenz
NO.165 : "Ja."
: "Was ist das?"
NO.167 : "In der Ank:undigung steht ein Thema, das maschinelle :Ubersetzung hei:st."
NO.168 : "Was f:ur ein Thema ist das?"
NO.169 : "Es tut mir sehr leid, fachliche Fragen kann ich nicht beantworten."
NO.170 : "In der zweiten Version der Ank:undigung stehen die Titel der Vortr:age, die auf der Konferenz gehalten werden."
NO.171 : "K:onnen Sie einen Blick hineinwerfen, bitte?"
NO.172 : "In Ordnung."
NO.173 : "Also gut, schicken Sie mir diese Ank:undigung so schnell wie m:oglich!"
: "Also gut, schicken Sie mir so schnell wie m:oglich diese Ank:undigung!"
NO.174 : "Mein Name ist Ahkira Watana-bee. Meine Adresse ist zwei, eins, einundsechzig, Schi-romi, Higaschi-ku, O:saka."
NO.175 : "Ihr Name ist Ahkira Watana-bee. Ihre Adresse ist zwei, eins, einundsechzig, Schi-romi, Higaschi-ku, O:saka, nicht wahr?"
NO.176 : "Ja."
NO.177 : "Ich werde Ihnen das sofort schicken."
: "Ich werde Ihnen sofort das schicken."
: "Gibt es sonst noch irgendetwas?"
NO.179 : "Nein."
NO.180 : "Es gibt nichts."
NO.181 : "Vielen Dank."
NO.182 : "Auf Wiederh:oren."
NO.183 : "Ja."
NO.184 : "Hier ist das Konferenzsekretariat."
NO.185 : "Ich m:ochte Sie etwas fragen."
NO.186 : "Ich m:ochte auf dieser Konferenz etwas vortragen."
NO.187 : "Wie soll ich vorgehen?"
NO.188 : "Bitte schicken Sie mir zuerst eine zweihundert W:orter lange Zusammenfassung bis zum zwanzigsten M:arz!"

NO.189 : "Ich begutachte das hier, und ich schicke Ihnen das Ergebnis bis zum zwanzigsten Mai."
NO.190 : "Ich schicke Manuskriptformulare mit, wenn der Beitrag akzeptiert worden ist."
NO.191 : "Bitte schicken Sie das Manuskript bis zum drei:sigsten Juni!"
NO.192 : "Ich verstehe."

NO.193 : "Auf was f:ur ein Formular soll ich die Zusammenfassung schreiben?"
NO.194 : "Es gibt ein spezielles Bewerbungsformular. Bitte f:ullen Sie das aus!"
NO.195 : "Also gut, ich schicke Ihnen ein Bewerbungsformular. K:onnte ich Ihre Adresse haben, bitte?"
NO.196 : "Ich verstehe."
NO.197 : "Das ist Ma-jumi Suhsuhki von ATR."
NO.198 : "Meine Adresse ist zwei, f:unf, drei, Higaschi-Ickehbuhkuro, Toschima-ku, Tokyo."
NO.199 : "Das ist Herr Ma-jumi Suhsuhki von ATR, nicht wahr?"
NO.200 : "Ihre Adresse ist zwei, f:unf, drei, Higaschi-Ickehbuhkuro, Toschima-ku, Tokyo, nicht wahr?"
NO.201 : "Ja."
NO.202 : "Das ist richtig."
NO.203 : "Also gut, schicken Sie ein Bewerbungsformular?"
NO.204 : "Ja."
NO.205 : "Ich verstehe."
NO.206 : "Also gut, ich schicke Ihnen das sofort."
: "Also gut, ich schicke Ihnen sofort das."
NO.207 : "Auf Wiederh:oren."
NO.208 : "Ist dort das Konferenzsekretariat?"
NO.209 : "Ja."
NO.210 : "Das ist das Konferenzsekretariat."
NO.211 : "Kann ich Ihnen helfen?"
NO.212 : "Ich m:ochte, da:s Sie mir sagen, wie ich zum Konferenzzentrum komme."
NO.213 : "Ich bin jetzt am Bahnhof Kyoto."
NO.214 : "Bitte fahren Sie mit der U-Bahn bis zum Bahnhof Kietahoh-dschi!"
NO.215 : "Von dort k:onnen Sie einen Bus zum internationalen Konferenzzentrum nehmen."
NO.216 : "Am Bahnhof Kietahoh-dschi k:onnen Sie ein Taxi nehmen."
NO.217 : "Wieviel kostet es, mit einem Taxi vom Bahnhof Kyoto zum Konferenzzentrum zu fahren?"
NO.218 : "Es kostet ungef:ahr sechstausend Jenn, wenn es ab dem Bahnhof Kyoto ist."
NO.219 : "Also gut, wieviel kostet es, wenn es ab dem Bahnhof Kietahoh-dschi ist?"
NO.220 : "Ungef:ahr neunhundert Jenn betr:agt es, wenn es ab dem Bahnhof Kietahoh-dschi ist."
NO.221 : "Ich verstehe."
NO.222 : "Vielen Dank."

NO.223 : "Nein."
NO.224 : "Gerne geschehen."
NO.225 : "Hallo?"
NO.226 : "Ja."
NO.227 : "Hier ist das Konferenzsekretariat.
NO.228 : "Ich m:ochte mich nach :Ubernachtungsm:oglichkeiten f:ur die Konferenz erkundigen."
NO.229 : "K:onnen Sie etwas vorschlagen, bitte?"
NO.230 : "Ja."
NO.231 : "Die Hotels, die wir vorschlagen k:onnen, sind das Kyoto-Hotel und das Kyoto-Prinz-Hotel."
NO.232 : "Der Preis f:ur ein Einzelzimmer betr:agt zwischen siebentausend und zehntausend Jenn pro Nacht."
NO.233 : "Der Preis f:ur ein Doppelzimmer betr:agt zwischen neuntausendf:unfhundert und sechzigtausend Jenn."
NO.234 : "Ich verstehe."
NO.235 : "Welches Hotel ist n:aher beim Konferenzzentrum?"
NO.236 : "Das Kyoto-Prinz-Hotel ist n:aher beim Konferenzzentrum."
NO.237 : "Ich m:ochte eine Reservierung f:ur das Kyoto-Prinz-Hotel vornehmen."
NO.238 : "K:onnten die Hotelbuchung Sie machen, bitte?"
NO.239 : "Ja."
NO.240 : "Ich kann eine Reservierung f:ur das Kyoto-Hotel und das Kyoto-Prinz-Hotel vornehmen."
NO.241 : "Ich verstehe."
NO.242 : "Also gut, kann ich ein Einzelzimmer f:ur siebentausend Jenn im Kyoto-Prinz-Hotel haben, bitte?"
: "Also gut, kann ich ein Einzelzimmer f:ur siebentausend Jenn in dem Kyoto-Prinz-Hotel haben, bitte?"
NO.243 : "Ja."
NO.244 : "Das ist ein Einzelzimmer f:ur siebentausend Jenn im Kyoto-Prinz-Hotel, nicht wahr?"
: "Das ist ein Einzelzimmer f:ur siebentausend Jenn in dem Kyoto-Prinz-Hotel, nicht wahr?"
NO.245 : "Ja."
NO.246 : "Das ist richtig."
NO.247 : "Ab wann sind Sie hier?"
NO.248 : "Das ist ab dem vierten August abends."
NO.249 : "Bis zum achten morgens, bitte!"
NO.250 : "Ich verstehe."
NO.251 : "Bitte warten Sie einen Augenblick!"
NO.252 : "Ich schaue nach, ob ich ein Zimmer bekommen kann."
NO.253 : "Ich kann ein Zimmer bekommen."

NO.254 : "Also gut, kann ich Ihren Namen und Ihre Adresse haben, bitte?"
NO.255 : "Das ist Kahsuhoh Nackamura."
NO.256 : "Meine Adresse ist eins, drei, eins, Schinnbaschi, Minnato-ku, Tokyo."
NO.257 : "Kann ich Ihre Telefonnummer haben, bitte?"
NO.258 : "Meine Telefonnummer ist drei, drei, eins, zwei, f:unf, zwei, eins."
NO.259 : "Ich verstehe."
NO.260 : "Ich habe ein Einzelzimmer im Kyoto-Prinz-Hotel vom vierten August bis zum achten bekommen."
NO.261 : "Vielen Dank."
NO.262 : "Auf Wiederh:oren."

References

[1] H. U. Block and S. Schachtl. Trace & unification grammar. In *Coling 92, Vol. 1*, pages 87–93, 1992.

[2] G. Gazdar, E. Klein, G. K. Pullum, and I. A. Sag. *Generalized Phrase Structure Grammar.* Blackwell, Oxford, 1985.

[3] T. Hasegawa. Feature structure rewriting system manual (revised version). Technical report TR–I–0187, ATR Interpreting Telephony Research Laboratories, Kyoto, 1990.

[4] G. Kikui. Feature structure based semantic head driven generation. In *Coling 92, Vol. 1*, pages 32–38, 1992.

[5] G. Kikui. Tree configurator for natural language generation. In *Proc. of Tag+ Workshop*, pages 482–486, U. Pennsylvania, June 1992.

[6] Y. Lepage. Easier C programming. Some useful objects. Technical report, ATR Interpreting Telephony Research Laboratories, Kyoto, November 1992.

[7] C. Pollard and I. A. Sag. *Information–Based Syntax and Semantics. Vol. 1: Fundamentals.* CSLI, Stanford, 1987.

[8] C. Pollard and I. A. Sag. *Information–Based Syntax and Semantics. Vol. 2: Topics in Binding and Control.* CSLI, Stanford, 1990. (draft).

[9] S. M. Shieber. *An Introduction to Unification–Based Approaches to Grammar.* CSLI, Stanford, 1986.

[10] M. Suzuki, G. Kikui, M. Seligman, H. S. Tropf, T. Morimoto, and A. Kurematsu. A Japanese–German speech translation system. In *Proc. IPSJ Spring Meeting, 1993*, Tokyo. (in Japanese; to appear).

[11] H. S. Tropf. Syntax in the spoken dialogue system SPICOS–II. In *Eurospeech 89, Vol. 1*, pages 30–33, Paris, September 1989.