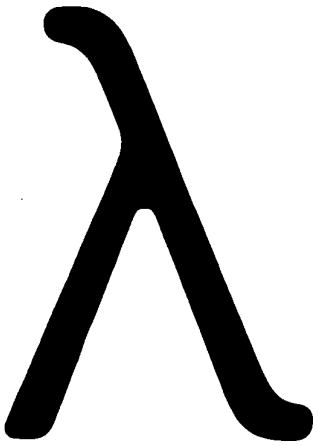
Lingvistik
Analyse
Metode
data **B**aser
Datamater
Applikation



Nr 7.

Nordiske Datalingvistikdage
og
Symposium for datamatstøttet leksikografi og terminologi
1987

Proceedings

Marts 1988

Institut for Datalingvistik
Handelshøjskolen i København

INDHOLDSFORTEGNELSE

Handelshøjskolen i København

PROGRAM

VI Nordiske Datalingvistikdage

3.-4.11 1987

Tirsdag 3.11 1987 Fabrikvej 7, lokale 428

8.30-9.00	Sekretariatet åbent for materialeudlevering, check-in, etc.
9.00-9.10	Steffen Leo Hansen: Velkomstord
9.10-9.50	Anna Sågvall Hein: "Leksikonet i en parser för svenska"
9.50-10.30	Gunnel Källgren: "What good is syntactic information in the lexicon of a syntactic parser?"
10.30-11.00	Kaffepause
11.00-11.40	Bengt Sigurd: "A referent grammatical analysis of some basic syntactic constructions"
11.40-12.20	Ole Togeby: "En syntaktisk parser på grundlag af Diderichsens feltskema"
12.20-13.40	Frokostpause
13.40-14.20	A. Lehtola & T. Honkela: "AWARE - Attributed Tree Transformations for Natural Language Interpretation"

14.20-15.00 K. Valkonen, T. Honkela, A. Lehtola: "Logical Trees as Canonical Representation of Query Sentences"

15.00-15.30 Kaffepause

15.30-16.10 Henrik Prebensen: "Diskursuniverser og nominalsyntagme-specificitet"

16.10-16.50 Lars Ahrenberg: "Ett system för objektinriktad dialog på naturlig svenska"

Onsdag 4.11 1987 Fabrikvej 7, lokale 428

9.00-9.40 Lars Borin: "A constraint-based approach to morphological analysis"

9.40-10.20 Torben Thrane: "Repræsentation"

10.20-10.50 Kaffepause

10.50-11.30 Benny Brodda: "Recent Developments of the Beta System"

11.30-12.10 Eva Ejerhed: "Processing text clause by clause"

12.10-13.30 Frokostpause

13.30-14.10 Frede Boje: "Disambiguering i human oversættelse versus maskinoversættelse"

14.10-14.50 Annelise Bech & Poul Andersen: "A Strategy for Disambiguation in a Multi-lingual MT-system"

14.50-15.20 Kaffepause

15.20-16.00 Klaus Schubert: "Att knyta Nordens språk till et mångspråkigt datoröversättningssystem"

16.00-16.40 Tove Fjeldvig & Anne Golden: "Bruk av språkbaserte hjelpemidler i informasjonssøkning"

16.40-17.00 Afslutning

PROGRAM

Datamatstøttet leksikografi og terminologi

5.-6.11 1987

Torsdag 5.11 1987

9.00-9.10	Åbning	
9.10-9.50	Gudrun Magnúsdottir: "Rapport fra 'Workshop on the Lexicon in Theoretical and Computational Perspective', 1987, Linguistic Institute, Stanford University"	Chairperson Henrik Holmboe
9.50-10.30	Lennart Lönngren: "Lexika, baserade på semantiske relationer"	
10.30-11.00	Kaffepause	
11.00-11.40	Ole Norling-Christensen: "Om læsning af maskinlæsbare tekster"	Chairperson K. Rossenbeck
11.40-12.20	Hanne Ruus: "Ordbøger i Danmark"	
12.20-13.40	Frokost	

Fredag 6.11 1987

9.00-9.40	Ivar Utne: "Terminologi - arbejds-instrukser - lagerstyring"	Chairperson
9.40-10.20	Bodil Nistrup Madsen: "Simulering af en relationel database"	G. Dyrberg
10.20-10.50	Kaffepause	
10.50-11.30	Krista Varantola: "Term banks, text banks and bank users"	Chairperson
11.30-12.10	Inge Gorm Hansen: "Erfaringer og resultater med Jurprojektet"	Birgitte Lauterbach
12.10-12.30	Afslutning	

Gunnel Källgren
Dep of Linguistics
Stockholm University
S-106 91 Stockholm, Sweden

What good is Syntactic Information in the Lexicon of a Syntactic Parser?

Imagine a situation where we want to parse texts and get as output information about the traditional grammatical categories of sentences, i. e. subject, direct and indirect object, finite verb, adverbials etc. That is an explicit goal of most parsers, and a necessary prerequisite for many applications of computational linguistics. How much information, syntactic and/or other, do we need in order to reach that goal?

In this paper I will look at the kind of syntactic information that is mostly given in syntactic parsers (sect. 1) and relate it to different parts of the parsing process (sect. 2). In section 3, I present an algorithm for assigning syntactic function to constituents without any use of lexical look-up. I discuss its results (sect. 4) and the type of errors that arise (sect. 5), which leads me to the conclusion that parsing can in most cases very well be done without a lexicon containing syntactic information. In section 6 I point to the need for different parsers for different purposes.

1. What kind of information can be given?

Many parsing systems use information that is centered around the verb. In the dictionary of the parser, each verb has information about whether it is transitive or intransitive and how obligatory or optional its complements are. Often some kind of selectional restrictions are also given. They describe properties of the referents of the noun phrases that enter different syntactical roles in relation to the verb. Such properties are concrete/abstract, animate/inanimate, human/non-human etc. The apparatus demands marking of the verbs with their functional frames, and marking of all words that can enter those frames accordingly, and procedures for checking the different kinds of marking against each other.

The basic thoughts behind this technique are clear; to get correct syntactic assignments in 1a-b and 2a-b one will need verbal frames like 3 and 4.

(1a) Mary gave the book to John.
(b) Mary gave John the book.
(2a) She drank the wine from a chrystal glass.
(b) The horse drank (the water) (in the bucket).

(3) give(x,y,z) where x=subj, y=dir obj, z=indir obj
patterns: x give y PREP z where PREP=to
x give z y

(4) drink(x,y,z) where x=subj, y=dir obj, z=loc advl
pattern: x drink (y) (PREP z) where PREP=from, in
selectional restrictions: x=animate, y=liquid, z=some kind of container

The classificatory system can then be developed and refined to cover more and more of the ways a word is used, and more and more of the vocabulary of a language (see Källgren 1986). Some existing systems use a much more finely graded classification, and some also include e g a classification of the adverbials that can modify a verb or a sentence. Fig. 5 shows two entries from a such system, the Janus system, that seems to be carefully planned and built on a larger scale than most others, see Cumming 1986 for a closer description of its design.

(5)

```
 :name 'LEAD
 :spelling "lead"
 :features '(VERB INFLECTABLE LEXICAL NOT-CASEPREPOSITIONS
OBJECTPERMITTED NOT-TOCOMP NOT-QUESTIONCOMP NOT-PARTICIPLECOMP
NOT-MAKECOMP NOT-BAREINFINITIVECOMP NOT-COPULA PASSIVE NOT-THATCOMP
NOT-ADJECTIVECOMP NONE-OF-BITRANSITIVE-INDIRECTOBJECT DOVERB DISPOSAL
EFFECTIVE OBJECTNOTREQUIRED NOT-OBJECTNOTPERMITTED SUBJECTCOMP
UNITARYSPELLING S-IRR PASTFORM EDPARTICIPLEFORM)
 :properties '((PASTFORM "led") (EDPARTICIPLEFORM "led"))

 :name 'SIMPLE
 :spelling "simple"
 :features '(ADJECTIVE NOT-CASEPREPOSITIONS R-ST DEGREE
COMPLEMENTPERMITTED TOCOMP FORNPPERMITTED SUBJECTHOLD NOT-SUBJECTCOMP
NONE-OF-APPROPRIATENESS-POSSIBILITYPROPERTY-OBVIOUSNESS NOT-THATCOMP
NOT-PREDICATEONLY)
 :properties '()
```

The lexicon must mainly be built up by hand and that is a task that takes good and clear linguistic intuitions to do. Actually, it is quite hard to imagine all the different kinds of complements that a verb can take. Sometimes subtle shades of meaning in the verb accompanies differences in the functional frame ('to read a book' but 'to read a child to sleep'). The verb can then be entered as two homonymous forms, or some other way of entering the information can be designed, but for the parsing process as such this does not necessarily remove all ambiguity. There is always the risk that some possibility is forgotten and the functional frame gets too small, or that some allowed pattern makes the frame too loose and unwanted constructions become allowed. In short, this way of putting conditions on the parse is hard, tedious and time-consuming without ever guaranteeing total correctness.

As a small excercise, the reader might want to try to figure out how the frame of the verb 'to read' should be described, given the corpus in 6, which is only a small subpart of the possible readings (!) of that single verb. What are objects and what are adverbials? When is a preposition not a preposition but an aspect marker? How can we capture that, in our culture, messages can be written on the side of buses, but not on trains or airplanes? And where precisely goes the borderline between signs that can be read and signs that can be read on?

(6a) She read the book.
(b) She read on the train.
(c) She read on the bus that there was a sale on. (Ambiguous.)
(d) She read the street sign to see where she was.
(e) *She read on the street sign to see where she was.
(f) She read on the sign that the store was closing at 6.
(g) *She read on the street sign that the street was 5th Ave.
(h) She read on without being disturbed by the noise.
(i) She read quickly.
(j) The book read quickly.
(k) She read the book quickly.
(l) The thermometer read 20 degrees.
(m) She read 20 degrees on the thermometer.
(n) Every morning she read the thermometer to see what to wear.
(o) She read the child to sleep.
(p) She read the book to pass the exam.
(q) She (read) (up on the mountain).
(r) She (read up) (on the mountain).
(s) She read German.
(t) She read law.

Some systems use an on-line ordinary dictionary instead. This makes large-scale parsing possible to an extent that will probably never be reached with the more handicraft based systems, but it brings with it the problem of extracting the wanted information from dictionary entries that were not written with this particular application in mind. The problem with full coverage of all possibilities also remains, but still this seems to be a more promising way. (Jensen and Binot 1987.)

2. Category assignment versus functional assignment

Parsing really consists of (at least) two parts, corresponding to the important difference between category and function. It is one task to identify the constituents of a sentence and assign their category, another task to decide the functional roles that the constituents play and to assign the structure of the whole sentence. Different languages signal categorial and functional information in different ways and to a different extent, a fact that must influence the mode of parsing chosen for a language. In a language like Finnish, both grammatical category and syntactic function can often be seen from the morphology of single words (Karlsson 1985), whereas in English, knowledge about overall sentence structure is often needed in order to assign the correct category to ambigous words. In 7, the two parsing levels clearly presuppose each other:

(7a) He judges sentence (7b) to be correct.
(b) Judges sentence criminals.

Swedish lies somewhere between Finnish and English in this respect. The morphology gives a good idea of the category of a word, especially when its close context is taken into account (Källgren 1984), while it is mainly the word order of the full sentence that gives information about the syntactic function of its constituents.

If category assignment and functional assignment are kept apart, what kind of information is needed for each of the two tasks? To identify constituents and their category, it is enough to know what

part of speech each word is and the allowed internal patterns of constituents such as noun phrases, prepositional phrases, and simple and complex verbs. If for the moment we disregard the problem of homonymy, we can generally say that a sequence of article + adjective + noun constitutes a noun phrase without knowing about the abstractness or animacy of the noun, and a finite verb is a finite verb, no matter what its functional frame may contain.

For the task of category assignment then, no syntactic information except part of speech and internal structure of certain phrasal categories seems to be needed. All the elaborate apparatus described above must thus be introduced in order to handle the assignment of syntactic function. This seems reasonable; the kind of conditions expressed in verb frames are conditions on sentence structure, not on constituent-internal structure. The purpose of adding the verb frames and the syntactico-semantic properties of the nouns is then to aid in the identification of subject, object, main verb etc. The question will thus be: Precisely what kind of information and how much information is needed in order to reach a good parse?

Tagging and parsing systems for English that make precisely this separation of tasks have been constructed for the large-scale analysis of the Brown and LOB corpora. Their systems for word-tagging, i e about the same as what was above called category assignment, are based on lexicon and morphology. TAGGIT is the system used for the Brown corpus. Its degree of correctness in deciding the grammatical class of words in unrestricted text is reported to be 77% (Greene and Rubin 1971). For CLAWS, the LOB system which is developed on the basis of TAGGIT, the degree of correctness is 96-97% (Garside 1987).

Independently of their work, I have built up a morphologically based system for category assignment in unrestricted Swedish text. For theoretical reasons, I have tried to limit the lexicon as much as possible, and today the system has a lexicon of less than 300 words that belong to closed categories or are morphologically highly irregular, which should be compared to the lexicon of 7 200 words in the CLAWS system. Except for those 300 words, all word-tagging in the Swedish system is based on morphological and contextual clues. As stated above, Swedish morphology contains more information and less ambiguous information than is the case with English, so the performance of the lexicon-less Swedish system is around 90% on any arbitrary text. The system is described in Källgren 1984a,b, 1985. It is written in BETA (see Brodda 1987) and runs on DEC-10/20 computers under TOPS-10/20 and is being implemented on PCs under DOS. Its output can give sentences that are analyzed like 8 and 9, or that simply look like the corresponding structures 10 and 11.

(8) S:(NP:(Mannen) Fin-V:(skrev) NP:(ett brev))
the man wrote a letter

(9) S:(NP:(Kvinnan) Fin-V:(visste) S:(Conj:(att) Pron-subj:(han)
the woman knew that he
Fin-V:(skrev) NP:(ett brev) Advl:(idag)))
wrote a letter today

(10) S:(NP Fin-V NP)
(11) S:(NP Fin-V S:(Conj Pron-subj Fin-V NP Advl))

3. An algorithm for functional assignment in Swedish sentences

To reach a full parse, I have constructed an algorithm for functional assignment in Swedish sentences, which is based on the output from the morphological system. The algorithm is now being implemented in BETA and CommonLisp. Its task is to decide, on the basis of as little information as in 10 and 11, the major syntactic roles of the constituents. I will here present the algorithm and its results and discuss the implications of it.

The algorithm has been applied to natural Swedish texts that have already been analyzed into sentences and clauses, and with the following major categories identified: noun phrases, pronouns in subject and object form, prepositional phrases, finite and non-finite verbs, auxiliaries, adverbs, prepositions, and conjunctions. As mentioned above, a system that does precisely this category assignment with a high degree of correctness (around 90%) already exists, but to be able to judge the output from the different parts of the parse separately, we have started from an idealized situation where the category assignment is taken to be 100% correct. Note that the algorithm does not presuppose correct text, it presupposes text that does not need to be corrected. (Cf. the discussion in section 6 about the different possible purposes of parsers.) This means that it gives a best-possible parse for every sentence, regardless of whether the sentence adheres to grammatical standards or not. A robustness of this kind is necessary to be able to deal with unrestricted text.

We regard each occurrence of a verb and its complements as a simplex sentence. The present version of the algorithm only works with clauses that contain a finite verb. Non-finite clauses pose an extra set of difficulties to any parser. Subordinate clauses get their analysis both as to the functional role they play in the superordinate sentence and as to their internal structure. The basic task is then to identify the subject of each simplex sentence and its direct and indirect object, in case any object(s) occur.

The algorithm is formulated as a set of rules (around 25 at present), of which some describe clear and unequivocal patterns and others give heuristic solutions for situations that can be ambiguous. Some examples:

A noun phrase "inside" a complex verb, i e between a finite and an infite verb, must always be the subject of the verb. Examples like 12a-b thus gives the rule 13.

(12a) Idag har mannen skrivit brevet.
today has the man written the letter
(b) Brevet har mannen skrivit idag.
the letter has the man written today

(13) X Fin-V NP Infin-V Y -> NP := subj

If the position before the finite verb (the so-called fundament position) is filled by an adverbial (an adverb or a prepositional phrase), the first noun phrase after the finite verb must be the subject. Examples 14a-b lead to rule 15.

(14a) Idag skrev mannen brevet.
today wrote the man the letter
(b) I arbetsrummet skrev mannen brevet.
in the study wrote the man the letter

(15a) Advl Fin-V NP X -> NP := subj
(b) PP Fin-V NP X -> NP := subj

Where no such rules are applicable, the heuristic simply says that the first noun phrase in a sentence is its subject. Rule 16 gives a correct result in sentences like 17a but not in 17b.

(16) NP Fin-V X -> NP := subj

(17a) Mannen skriver alla brev i arbetsrummet.
the man writes all letters in the study
(b) Alla brev skriver mannen i arbetsrummet.
all letters writes the man in the study

A set of analogous rules decide the assignment of direct object (mostly simply the noun phrase that is not subject) and the choice between direct and indirect object where two objects occur.

4. Some results of the application of the algorithm

The table below (Fig 18) gives the results for a corpus of 1,451 simplex sentences, taken from different text types. Of the 1,451 simplex sentences, 160 lack a finite verb, so the figures in the table are computed on the basis of the 1,291 sentences that the algorithm applies to.

Instances where the algorithm (correctly) predicts zero subject, as in imperatives, count as a correct identification of subject. For the identification of objects, two figures are given. The first covers the cases where object(s) occur without a preposition in the sentence, while the second also includes instances with particle verbs ('skriva ner något' write down something) and instances where a prepositional phrase can on semantic grounds be regarded as a direct or indirect object. The last category can be difficult to judge, but some clear instances exist ('titta på TV' look at TV, 'ge boken till flickan' give the book to the girl, as compared to 'ge flickan boken' give the girl the book). Generally, a human linguist in many cases has quite a hard time in choosing between particle + object/prepositional object/prepositional adverbial phrase. There are often no clear rules-of-thumb and different people come to different decisions, so the resulting figures in themselves are not that important. The important thing is that such instances occur, and that a weak point of the algorithm is its inability to decide the syntactic function of prepositional phrases. They are at present all regarded as some unspecified kind of adverbials. However, this deficiency is not as destructive as one might fear, since there are in fact not that many such occurrences.

(18) Ratio of correct functional assignments

	%	N
Correct subject/total number of sentences	99,5	1 284/1 291
Correct dir object/"naked" dir objects	98,6	579/587
Correct dir obj/"naked" + prep dir obj	83,5	579/693
Correct indir obj/"naked" indir obj	100	3/3
Correct indir obj/"naked" + prep indir obj	60	3/5

The ratios for identification of subject and direct and indirect object without preposition are remarkably high. Indirect objects are surprisingly infrequent. In the whole material of almost 1,300 sentences only five indirect objects appear, three without a preposition that are all correctly identified and two instances with a preposition that have been missed by the algorithm. The really disturbing figure is the one for direct objects with preposition. The algorithm has missed 106 such objects, in total 15% of all direct objects, plus 8 direct objects without a preposition. Most of the latter errors are instances where the algorithm mixes up subject and direct object (cf 17b above). The number of such errors is however low, considering the fact that Swedish often has OSV order for reasons of textual coherence.

5. Some analysis errors occuring in the material

However, the most interesting results of the application of the algorithm are perhaps the errors. What type of structures can this simplistic method not manage? Can it be improved and developed? Would more elaborate systems like those mentioned in section 1 do better? What kind of information - syntactic, semantic, pragmatic - would be needed in a lexically based parser? I will here give one example of every error type occuring in the material and discuss what kind of extra information would 'rescue' them.

The examples below are all chosen because they are typical for the kind of errors they represent. As can be seen, many sentences are quite weird and some are even ungrammatical, but that is not an argument for excluding them from the corpus. The aim of my algorithm is to handle unrestricted text; the strange sentences have occured in normal texts and should thus be given an analysis. (Cf section 6.) As a matter of fact, there are also ungrammatical sentences that get correct functional assignment by the algorithm.

Below each example I give the correct analysis and an English translation. As a comparison I then show a structurally analogous sentence, where the algorithm's analysis would be the correct one. It is clear that in most cases the latter sentence represents a more common pattern.

Erroneous subject:

(19) Plötsligt, mitt på den sterila slätten, buktar gräsmattor
(Adverbial) FV1 Subj of FV1
och står en byggnad som ser ut som ett rymdskepp.
Conj FV2 (Subj of FV2)

'Suddenly, on the sterile plain, bend lawns
and stands a building that looks like a space ship.'

Analyzed in analogy with:

(20) På slätten buktar gräsmattor och pryder sin omgivning.

'On the plain bend lawns and adorn their surroundings.'

The algorithm says that if a finite verb comes immediately after a sentence conjunction, the subject of the preceding clause is the subject of that verb. (There is no number agreement between subject and verb in Swedish.) The error in 19 would be solved by the extra information that 'står' (stands) is intransitive, so that 'en byggnad...' cannot be a direct object, and, for semantic reasons (however they are to be specified), cannot be a temporal or locative adverbial; thus it must be the subject. At the same time I find it hard to believe that the selectional restrictions of a carefully designed lexicon would allow lawns to be the subject of the verb bend.

Mixing of subject and direct object:

(21) Större betydelse än riksdagen har under hösten
(Object?) FV (Adverbial)
två andra processer.
(Subject)

'Greater importance than the parliament have during autumn two other processes.'

Analyzed in analogy with:

(22) Större universitet än Linköping har under hösten två andra problem.

'Larger universities than L. have during autumn two other problems.'

Sentence 21 has a verb phrase that is as problematic to a human as to a computer. Is the main verb a simply 'har' (has) with a nominal direct object, or is it 'har ... betydelse (än)' (is of ... importance (than))? Most analyzers would probably prefer the second solution, which would have to be given as an entity in a lexicon, with slot for comparative adjective and all. But the situation is then messed up further by the splitting and fronting of the last part of the phrase. It is highly unlikely than any lexical look-up mechanism would be able to restore this discontinuous constituent.

Erroneus direct object:

(23) Jag glömmer aldrig när Palme var i Frankrike.
Subj FV Advl (Object)

'I never forget when Palme was in France.'

Analyzed in analogy with:

(24) Jag glömmer aldrig när jag skriver upp saker i almanackan.

'I never forget when I write things in my calendar.'

On rare occasions, a temporal clause can appear as a direct object of some verbs. Those verbs can however also take temporal clauses as adverbials. I can see no generalizable way of telling when the when-clause is an adverbial and when it is an object in sentences like 23 and 24. The adverbial reading must have an overwhelming probability in its favour.

Erroneous direct object with complex verb:

(25) ... när den ena fick tag i ett snöre.
(Subj) (FV) (Object)

'... when one got hold of a string.'

Analyzed in analogy with:

(26) ... när den ena fick buggar i ett program.

'... when one got bugs in a program.'

The main verb is the whole phrase 'fick tag i' (got hold of) with 'i' as a verbal particle, but it is interpreted with the noun 'tag' as direct object and 'i' as a preposition starting a prepositional phrase incorporating what should really be the direct object. All those complex verbs - and there are many of them - must be listed as idioms, but there remains the problem that some of them can have literal readings as well.

Erroneous indirect object:

No instances in the material.

Missed prepositional direct object:

(27) ... det avvek från vad parterna träffat förlikning om, ...
Subj FV (Object)

'... it deviated from what the parties had settled, ...'

Analyzed in analogy with:

(28) Han avvek från anstalten.

'He deviated from the prison.' = 'He escaped.'

In its concrete meaning, a verb like 'avvek' is constructed with a locative adverbial, but it can also have a transferred meaning, where the locative reading of the complement seems less natural and an object interpretation is closer at hand. In these cases, there are mostly clear instances at either end and a grey zone in the middle.

Missed prepositional indirect object:

(29) (Det vet man väl) vad han gör med oss.
Obj Subj FV (Indir Obj)

'(One sure knows) what he does to us.'

Analyzed in analogy with:

(30) Det vet man väl vad han gör med kniven.

'One sure knows what he does with the knife.'

The particle verb 'gör med' with a sense of doing something against somebody/something has a particle identical to the instrumental preposition. A lexicon might tell us that knives are typical instruments and persons are not, and thus guide the analyses of 29 and 30; but how would it handle a sentence like 'One sure knows what he does with them'? A full anaphora resolution is necessary to decide whether the antecedent of 'them' is animate or not and, consequently, whether 'them' is to be analyzed as indirect object or instrumental adverbial. This does not always help either: 'books' are certainly not instrument in 'What will you do with all your books when you move?'

Many of the errors (21, 25, 27, 29) exemplify a general tendency. Verbs are often used with a transferred meaning that almost always implies an abstraction in comparison to their 'prototypical' meaning, or the meaning of the predicate is given by an 'empty' verb (be, do, have, get) plus a complement. In both cases, the pattern of functional roles of the verb/predicate is often affected.

Thus, from all this we can conclude that most of the functional assignment can be done without any recourse to lexicon at all, and for those cases that remain, the information necessary for a correct assignment is of a complex and often rather dubious nature. From the error analysis and from the senteces with 'read' in ex. 6, we can see that it is an enormous task to think of all the possible constructions, to find minimal properties that are not too ad hoc and that can allow the desired sentences and exclude the others, and to build a system that can check all these possibilities and keep track of what it is doing. To get both readings of 6c for instance, we would have to build into a parser the knowledge of the world that (in this culture) buses, but not trains or airplanes, can be used as moving signboards for advertising, and that it is thus possible either to sit on a bus and read, or to stand beside it and read on it. We must ask ourselves if this is the kind of information we would want to have in a syntactic parser and we must be aware of the fact that there will always be examples that we cannot manage. We must also acknowledge the fact that different purposes have different demands on the performance of parsers.

6. Parsers adjusted to different purposes

Parsing is no longer only a research enterprise in itself, but will more and more become a necessary prerequisite for other kinds of theoretical research or practical applications, and that fact must influence the choice of parsing algorithm. The important question of what to do with ungrammatical input should also depend on the purpose of the parser.

A parser that is used to test a specific linguistic theory should reject all input that is not in accordance with what the theory predicts. A broad coverage of language is not seen as very

important and there is a clear distinction between sentences that the parser can and can not handle, a distinction which not always corresponds to the intuitions of a human language user. Ungrammatical sentences are rejected, mostly without indication of what caused their ungrammaticality.

A parser that is part of a grammar check or critiquing system of some editor should note all deviant sentences and perhaps even suggest how they should be corrected. It can also note stylistic features and compute frequencies for single words as well as constructions. Such parsers are supposed to cover large parts of the language and to find and diagnose literally all instances of ungrammaticality. This makes their task considerably more difficult than that of the model testers, and truly there are no fullfledged critiquing systems available at present even if some attempts seem promising.

A parser that is to be used in connection with e g a question-answering system must have the ability to decide not only what is grammatical but also what makes sense. It will need more semantics than the other parsers, often in the form of knowledge about the database to which it is connected. Questions of grammaticality can be of less importance as long as the input is interpretable.

The work with large-scale corpora that is emerging in today's linguistics demands of a parser that it can analyze sentences, add structural information, and always give a best-possible analysis, with or without at the same time signalling if there is something wrong with the sentence. Its input is more or less unrestricted text and its output can be used for many forms of linguistic research, also of the model-testing kind, as well as for building and updating databases etc. In this connection there are many purposes for which a fast, simple, and robust parser of the kind suggested here is precisely what is needed.

References:

Brodda, Benny, 1987. Tracing Turns in the London-Lund Corpus with BetaText. In: Proceedings from the ALLC conference, Goteborg June 1987. Forthcoming, Sprakdata, Goteborg University.

Garside, Roger, 1987. The CLAWS wordtagging system. In: Garside - Leech - Sampson, The Computational Analysis of English. London: Longman.

Green, B.B. - Rubin, G.M., 1971. Automatic grammatical tagging of English. Providence, R.I.: Department of Linguistics, Brown University.

Jensen, Karen - Binot, Jean-Louis, 1986. Disambiguating prepositional phrase attachments by using on-line dictionary definitions. IBM Research Report RC 12148, Yorktown Heights.

Källgren, Gunnel, 1984a. HP - A Heuristic Finite State Parser Based on Morphology. In: Sågvall-Hein, Anna (ed.) De nordiska datalingvistikdagarna 1983, p. 155-162. Uppsala universitet 1984.

Källgren, Gunnel. 1984b. Automatisk excerpering av substantiv ur löpande text. Ett möjligt hjälpmedel vid automatisk indexering? IRI-rapport 1984:1. Institutet för Rättsinformatik, Stockholms universitet.

Källgren, Gunnel, 1985. A Pattern Matching Parser. In: Togeby, O. Papers from the Eight Scandinavian Conference of Linguistics. Copenhagen University.

Källgren, Gunnel. 1986. The Role of Lexicon in Computerized Analysis of Natural Language. In Dahl (ed.) Papers from the Ninth Scandinavian Conference of Linguistics. Stockholm University.

Karlsson, Fred, 1985 (ed.). Computational Morphosyntax. Report on Research 1981-1984. University of Helsinki, Publications No. 13 1985. Helsinki.

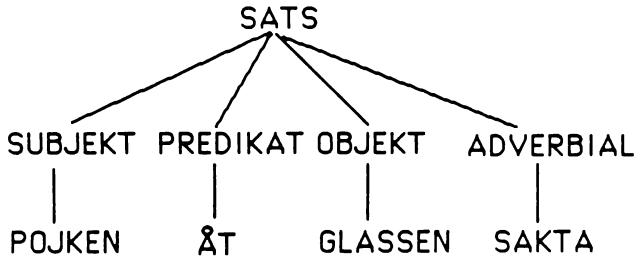
BENGT SIGURD
Inst för Fonetik och Lingvistik, LUNDS Universitet
Helgonabacken 12, S-22362 LUND

REFERENTGRAMMATIK - EN KORT PRESENTATION

Satsdelsanalys (funktionell analys)

I analys av meningar och satser enligt traditionell skolgrammatik skiljde man mellan identifiering av ordklasser, t.ex. substantiv, verb, adverb och identifieringen av satsdelar, t.ex. subjekt, predikat, objekt, adverbial. I satsen "Pojken åt glassen sakta" skulle man enligt klassisk skolgrammatik säga att "pojken" och "glassen" är substantiv, "åt" är verb och "sakta" är adverb. Om man ombads ta ut satsdelarna skulle man säga att "pojken" är subjekt, "åt" är predikat, "glassen" är objekt och "sakta" är adverbial. Om man skulle rita ett diagram var det fråga om ett satsdelsdiagram och det kunde då bli som nedan där en motsvarande parentesnotation också givits.

Funktionell analys (satsdelsanalys)
Trädrepresentation



Parentesrepresentation
s(subj(pojken),pred(åt),obj(glassen),advl(sakta))

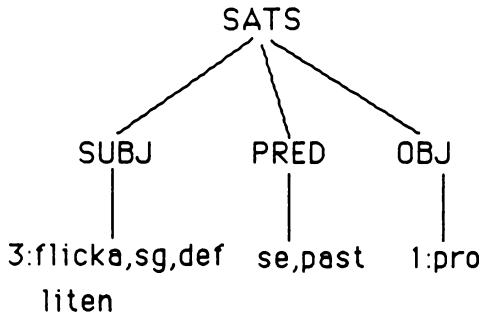
Referentgrammatik gör samma uppdelning i en ordklassanalys och en satsdelsanalys, men kallar den senare funktionell analys i anslutning till modernt internationellt språkbruk. Varken traditionell eller modernare grammatikforskning har givit terminologi och metodik som gör att man är överens i alla detaljer om hur olika ord och ordgrupper skall benämnas. När man utvecklar en grammatik som skall kunna tillämpas av en dator måste man

emellertid bestämma sig för terminologi och representationssätt och vara konsekvent. Vi skall beröra några av dessa problem.

I parentesnotationen ovan har satsdelsbeteckningarna förkortats och bl.a. dessa förkortningar återkommer i bilagda utskrifter av datorkörningar: s=sats, subj=subjekt, obj=objekt, pred=predikat, advl=adverbial, sadvl=satsadverbial.

De funktionella representationerna är avsedda att vara ett interface mot logik och semantik. Att döma av den framgång man haft när man beskrivit språk efter denna traditionella mall har de en avsevärd universalitet. Det är det som gör det möjligt att använda dem som en mellanrepresentation vid översättning mellan språk. Det är en fördel om ett standardformat då kan användas och i den datorimplementering av referentgrammatik som gjorts av SWETRA (Swedish Automatic Translation Group, Lund) har man preliminärt bestämt sig för ett format omfattande maximalt 10 konstituenter (också nämnda i denna ordning i funktionella representationer): subj, pradv, pred, dobj, obj, sadvl, sadvl, advl, advl, advl. I följande sats förekommer t.ex. ett predikat, ett partikeladverbial (pradv), två satsadverbial samt två vanliga adverbial: "Pojken kanske inte sprang in till staden igår. Motsvarande funktionella representation skulle bli: s(subj(pojken),pradv(in), pred(sprang), sadvl(kanske),sadvl(inte), advl(till,staden),advl(igår)).

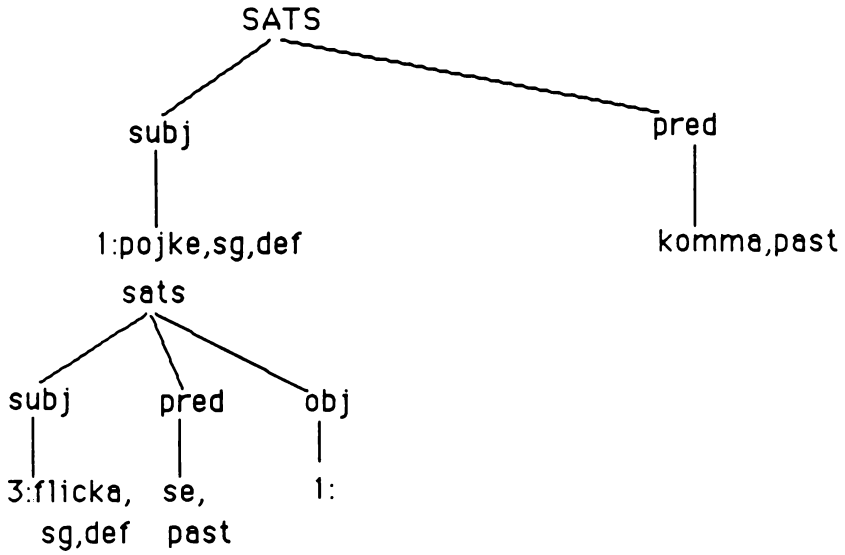
Bestämningar till substantiv inom en nominalfras kan föras upp (efter "nom" =namn i datorprogrammet) som ytterligare upplysningar efter huvudordet, det ord som avgör kongruensen inom nominalfrasen. Man kan också analysera orden vidare morfologiskt och t.ex. sätta "pojke,sg,best" för "pojken" eller, med användande av angliserande semantisk representation "boy,sg,def". På samma sätt kan man representera "åt" med "eat, past". I datorimplementeringar av referentgrammatik avsedda för automatisk översättning användes normalt sådana anglicerande semantiska representationer (machinese). En funktionell analys av "Den lilla flickan såg honom" är följande.



s(subj(3:flicka,sg,def,liten),pred(se,past),obj(1:pro))

I denna funktionella representation har också satts in de referentnummer som givit referentgrammatik dess namn. Subjektet räknas som referent nummer 3 i den text som de två exempelmeningarna bildar: "Pojken(1) åt glassen(2) sakta. Den lilla flickan(3) såg honom(1)". Objektet i den sista satsen har fått nummer 1, eftersom det är fråga om samma referent som tidigare, vilken hänvisas till med ett pronomen (pro) denna gång.

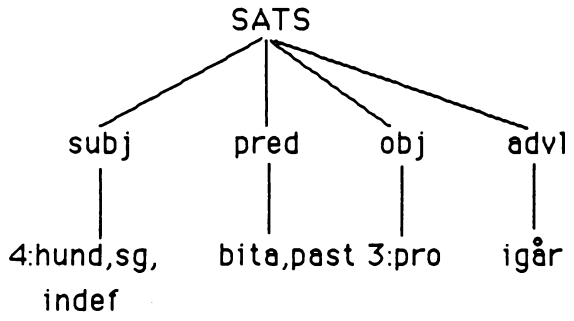
En relativsats är också en bestämning inom en nominalfras och den kan ge en ytterligare upplysning om referenten ifråga. En funktionell analys av "Pojken som flickan såg kom" är följande:



s(subj(1:pojke,s(subj(3:flicka),pred(se,past),obj(1))),pred(kom))
sg,def sg,def

Dessa representationer visar att bland bestämningarna till referent nummer 1 finns en sats där den tidigare nämnda flickan (referent nummer 3) är subjekt och referent nummer 1 objekt. Detta stämmer väl med hur man brukar uttrycka sig i fråga om relativsatser. Man skulle här säga att objektet i relativsatsen "som flickan såg" är "som", vilket i sin tur syftar på samma sak som ordet "pojken" (korrelatet). Referentgrammatik konkretiserar genom sina nummer den referent som traditionell och modern grammatik talar om.

Satsen "Igår bet en hund henne" har nedanstående funktionella representation, där ordningen i den funktionella representationen blir: subj, pred, obj, advl, även om satsdelarna inte kommer i den ordningen i den föreliggande satsen. Ordet "henne" antas syfta på den tidigare nämnda flickan och hunden blir då den fjärde (nominella) referenten i texten.



s(subj(4:hund,sg,indef),pred(bita,past),obj(3:pro),advl(igår))

Referentgrammatisk analys ger också uppgift om satsens modus (deklarativ, fråga, imperativ) och fokuserad satsdel - i satsen ovan "igår". Satser kan ha samma funktionella representationer, men olika satsdel i fokus (som fundament för att följa Diderichsens terminologi). Funktionella representationer för några satser med s.k. formellt subjekt ("det") är följande, där "_" markerar tom plats:

Funktionell representation	*Sats*
s(subj(_),pred(regna,past))	Det regnar
s(subj(sparv),pred(sitter),advl(där))	Det sitter en sparv där
s(subj(s(subj(du),pred(kom)),pred(bra))	Det var bra att du kom

Man kan själv välja hur detaljerad eller specifik man vill ha sin representation: om man vill sätta ut referentsiffror, om man vill ge detaljerad ordanalys etc. I satsen "Det var bra att du kom" har inte tempus (past) representerats (vilket man kan kan göra genom att tillägga "past" vid "bra"). Den motsvarande satsen "Att du kom var bra" har samma funktionella representation i referentgrammatik, men i den är fokuseringen en annan. Satsen "En sparv sitter där" skulle få samma funktionella representation som "Det sitter en sparv där", men har inte samma konstituent i fokus. Många småord t.ex. "som", "att", "det" syns inte i den funktionella representationen.

Man kan utforma den funktionella representationen med tanke på de syften man har med den. Det är naturligt att låta frågor få en funktionell representation som motsvarar svaret. Det gör att den funktionella representationen motsvarande "Vem bet hunden igår?" bör vara: s(subj(vem),pred(bet),obj(hunden),advl(igår)) och den funktionella representationen motsvarande "När bet hunden henne?" bör vara: s(subj(hunden),pred(bet),obj(henne),advl(när)). Frågan "Bet hunden henne igår?" kan ha samma funktionella representation som "Hunden bet henne igår"; skillnaden noteras i modusvärdet som är "d"(=deklarativ)" för ett påstående, "q"(=question) för en fråga. I frågor med frågeord anses de stå i fokus, i ja/nej-frågor anses det finita verbet stå i fokus. Referentgrammatik är förberedd för textlingvistisk analys genom möjligheten att hålla reda på referenterna (diskursreferenterna) i texten och registrera vad som fokuseras i serier av meningar i en text.

En funktionell representation för frågan "Vem sade pojken att hunden bet?" bör identifiera "vem" som objekt till "bet", men dessutom bör den ta hänsyn till att en frågan "Vad sade pojken?" bör ha "vad" som objekt och att att-satsen bör vara objekt i svaret "Pojken sade att en hund bet flickan". En lämplig funktionell representation för "Vem sade pojken att hunden bet?" är då följande: s(subj(pojken),pred(sade),obj(s(subj(hunden),pred(bet),obj(vem)))

Liksom ord t.ex. "som" kan sakna motsvarighet i den funktionella representationen så kan den funktionella representationen innehålla element som saknas i den föreliggande satsen. Exempel på detta ger s.k. kontrollverb, t.ex. "lova". Satsen "Per lovade (att) komma" brukar analyseras så att man säger att subjektet i den överordnade satsen (Per) kontrollerar och är subjekt till infinitiven "komma". I

referentgrammatik kan man visa detta i den funktionella representationen genom att sätta in ett subjekt vid infinitiven: s(subj(Per),pred(lovade),obj(s(subj(Per),pred(komma)))

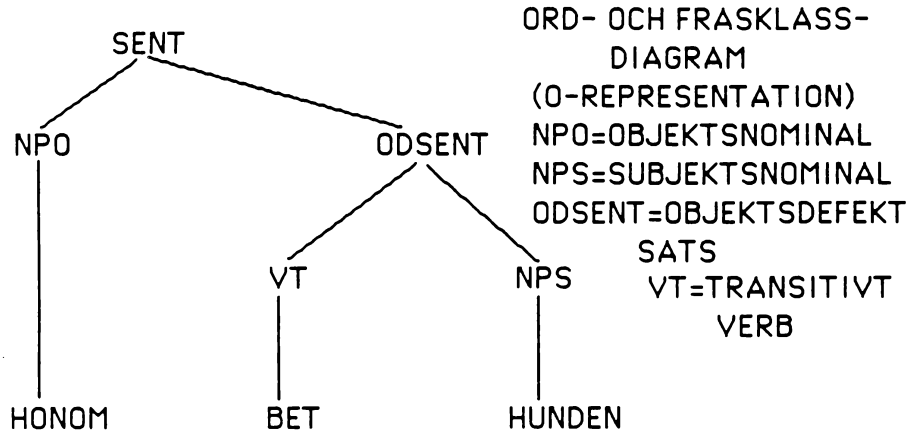
Ord- och frasklassanalys (kategorianalys)

Som nämnts skiljer referentgrammatik liksom traditionell grammatik mellan funktionell analys i subjekt, predikat, objekt etc. och analys i ordklasser. Det finns inte en så väletablerad modern term som funktionell analys för ordklassanalys och vi väljer att tala om ord- och frasklassanalys, eftersom analysen utvidgas till fraser. Vi räknar med både fulla och defekta fraser i referentgrammatik.

Uppslaget att räkna med defekta fraser kommer från Gazdar. En kategori (fras eller sats) säges vara defekt därför att den saknar en "väntad" konstituent. En prepositionsfras som "på båten" blir defekt då den bara innehåller prepositionen "på" som i satsen: "Vad tittade Per på?" (Uppenbarligen är "Vad" den saknade konstituenten). I satsen "Flickan trodde pojken att hunden bet?" är bisatsen "att hunden bet" defekt därför att det transitiva verbet "bet" saknar ett objekt - det som motsvaras av "Flickan". I den funktionella representationen är dessa saknade element placerade på sina platser såsom vi har sett. Gazdar intresserar sig i sin grammatik, som kallas Generaliserad frasstrukturgrammatik, dock bara för vad vi kallar ord- och frasklassanalys inte för en särskild funktionell representation. Referentgrammatik har övertagit en rad benämningar för ord- och fraskategorier och försökt systematisera dem så att de kan användas i datorprogram för analys och syntes av meningar. Kategorierna väljs och definieras så att de kan användas i de program av grammatiska regler som härleder en funktionell representation från en föreliggande sats.

Meningen "Honom bet hunden" har klart en funktionell representation: s(subj(hunden),pred(bet),obj(honom)) och en analys enligt referentgrammatikens principer ger också upplysningen att det är en deklarativ sats och att "honom" står i fokus. En referentgrammatik för svenska måste skilja mellan subjektsnominalfraser (nps) och objektsnominalfraser (npo) även om bara pronomen visar denna skillnad genom skilda former. I den föreliggande meningen identifierar vi "honom" som ett objektspronomen (proo i internationell terminologi), "bet" som ett

transitivt finit verb (vt) och "hunden" som en form som kan stå som subjektsnominal (nps). Tillsammans utgör de sista två orden "bet hunden" en objektsdefekt sats och vi benämner en sådan sats "odsent" där o=objekt, d=defekt,sent=sentence. Vi konstaterar att en objektsdefekt sats innehåller ett transitivt verb följt av ett nps. En motsvarande ord- och frasklassrepresentation (o-representation) i parentesform blir då: sent(npo(honom),odsent(vt(bet),nps(hunden))). Vi använder "sent" som namn på rotnoden i o-representationen. Detta motsvarar nedanstående träddiagram under vilket vi också skrivit den funktionella representationen.



Funktionell representation
s(subj(hunden),pred(bet),objekt(honom))

Ord- och frasklassanalysen kan ses nerifrån (motsvarande en bottom-up-process) som identifikation av "honom" som ett NPO, av "bet" som ett VT och "hunden" som ett NPS. Ett VT följt av ett NPS kan sedan enligt en regel identifieras som en ODSENT och ett NPO följt av en ODSENT kan identifieras som en SENT. Sett uppifrån (som en top-down-process) kan diagrammet sägas visa att en sats (SENT) kan bestå av ett objektsnominal (NPO) om det följer en objektsdefekt sats (ODSENT) efter. En objektsdefekt sats kan bestå av ett transitivt verb (VT) följt av ett subjektsnominal (NPS).

I satsen "Vem bet hunden?" kan vi på motsvarande sätt se "Vem" som ett objektsnominal (ett frågeobjektsnominal: npqo) följt av en sdsent, om vi tänker på en sats med den funktionella representationen: s(subj(hunden),pred(bet),obj(vem)). Samma sats har uppenbarligen en annan möjlig -om än oväntad- tolkning

motsvarande: s(subj(vem),pred(bet),hunden)) . Den senare tolkningen motsvarar o-analysen: sent(nps(vem),sdsent(vt(bet),npo(hunden)). Här betecknar "sdsent" subjektsdefekt sats - vi skall se närmare på de olika kategorierna nedan.

Referentgrammatiska regler

De grammatiska reglerna i referentgrammatik beskriver hur olika kategorier kan kombineras till högre kategorier, vilken ordning kategorier skall komma i och vilka ord som tillhör dem. Dessutom talar de referentgrammatiska reglerna om hur den motsvarande funktionella representationen ser ut. Reglerna härleder den funktionella representationen samtidigt som de tillämpas på en föreliggande sats. Omvänt kan reglerna härleda de sekvenser av fraskategorier och slutligen ord som motsvarar en funktionell representation. Regler för referentgrammatik skrivs i ett för lingvister bekvämt format kallat DCG (Definite Clause Grammar) som vanligen finns tillgängligt i nyare implementeringar av programmeringsspråket Prolog. Sådana regler förstår datorn direkt och reglerna kan alltså användas både för analys (parsning) och syntes (generering) av meningar. De regler som behövs för att göra de ovan beskrivna analyserna av satsen "Honom bet hunden" är följande (något förenklade, bl.a. utan referentnummer; "_" kan stå på en plats för en variabel som inte är relevant (anonym) för tillfället):

```
sent(d,_,X,F) --> npo(X),odsent(_,_,X,F).
odsent(_,_,X,s(subj(Y),pred(Z),obj(X)) --> vt(Z),nps(Y).
npo(honom) --> [honom].
nps(hunden) --> [hunden].
vt(bet) --> [bet].
```

Om man skriver: sent(M,_,T,F,[honom,bet,hunden],[]) till datorn (vederbörligen laddad) så returnerar den den önskade funktionella representationen såsom ett värde hos F och "d" som ett värde för modus (M). Reglerna ger inte någon o-representation. Om man skriver sent(d,_,honom,s(subj(hunden),pred(bet),obj(honom)),X,[]) så ger programmet satsen "honom,bet,hunden" som ett värde för X. ([] - den tomma listan - kan sägas uttrycka att det inte blir några ord kvar).

I reglerna, liksom allmänt i Prolog, användes stora bokstäver för att beteckna variabler. Man kan beskriva hur reglerna tillämpas i analys (parsning) på följande sätt. Se efter om det första ordet (X) är en npo. Om så är fallet se efter om det kommer en odsent efter. Ta i så fall dess funktionella representation (F) och sätt på motsvarande plats i sent. Odsent saknar en konstituent och kommer att sätta in det funna X som objekt i sin funktionella representation. Den funktionella representation som hamnar på plats i sent kommer därför att vara komplett. I den aktuella meningen hittar programmet "bet" som är ett transitivt verb (vt) och blir värdet för Z. Sist hittar programmet "hunden" som är ett nps och då blir Y. Värdena av X,Y,Z sätts då in i den funktionella representationen till vänster om pilen. Prolog har en inbyggd parser som försöker utföra den uppgift man beskrivit i reglerna. Prolog kallas ett deklarativt språk (i motsats till procedurala språk). Man kan emellertid följa sökprocessens gång genom att använda inbyggda kommandon som "trace" och "spy".

Lägg märke till att referentgrammatik inte behöver några transformationsregler. Formalismen tillåter en att specificera vilken representation man vill till vänster om pilen som en motsvarighet till ord- och fraskategorierna till höger om pilen. Man kan placera satsdelarna i vilken ordning man vill och kalla dem vad man vill. Likaså behöver inte referentgrammatik några spår (trace, e) eller noll-konstituenter som EST-, GB- och GPSG-grammatik. Flyttade konstituenter kan sägas åka med på vissa specialplatser i fraskategorierna såsom framgått. Dessa specialplatser fungerar som tillfälliga landningsplatser och motsvarar i vissa fall de COMP-noder som andra modeller utnyttjar för liknande ändamål (se Sell,1986).

Vi skall emellertid inte gå in längre på reglerna här, men tala om att de såvitt vi kan se kan hantera alla de problem som behöver hanteras: ordföljd, optionella konstituenter, kongruens, långa flyttningar (unbounded dependencies), ordböjning. Det finns ganska omfattande referentgrammatiska regler (program) för svenska och engelska som styrker detta påstående. Dessa moduler är utarbetade inom SWETRA där de används för översättning mellan språken via den funktionella representationen.

Mening, huvudsats och bisats

Vi har hittills talat om satser och meningar som man brukar utan att göra klar åtskillnad. I själva verket måste man betrakta mening (meng) som en högre enhet som kan bestå av en eller flera huvudsatser (sent) vilka i sin tur kan innehålla en eller flera bisatser (sunt) som satsdelar - i regel objekt eller adverbial. Vi räknar med att det egentligen är meningen som innehåller interpunktionstecknet (punkt, frågetecken,utropstecken). Meningen "Men pojken sprang och flickan hoppade." anses ha följande o-representation:

meng(konj(men),s(subj(pojken),pred(sprang),konj(och),
s(subj(flickan),pred(hoppade))).

Vi kallar "men" och "och" för konjunktioner i anslutning till traditionell grammatik här. Lägg märke till att den inledande konjunktionen "men" inte leder till omvänd ordföljd i den efterföljande satsen vilket ett inledande adverb i en sats (sent) skulle ha gjort. Samordnade satser där subjektet i den senare satsen saknas (strukits, eliderats) kan i referentgrammatik naturligen beskrivas genom att den senare satsen betraktas som subjektsdefekt. Så kan man t.ex. analysera: "Pojken sprang och sjöng" där "sjöng" då representerar den subjektsdefekta satsen, men man kan i många fall tveka om man skall analysera en sådan sats såsom innehållande två samordnade verb i stället.

Det är nödvändigt att skilja mellan huvudsats (sent) och bisats (sunt) i svenskan (däremot inte i t.ex. engelskan och polskan), eftersom som bekant satsadverben placerar sig efter det finita verbet i huvudsats men före i bisats. Vi illustrerar de olika ordningarna av konstituenter som karakteriserar olika satstyper i appendix. Där visar vi också hur man generellt kan se svenska satser såsom bestående av ett initialt element följt av en konstituent som saknar detta element. Man kan betrakta "Honom bet hunden" som ett npo(honom) följt av en odsent(vt(bet),sps(hunden)). Men vi betraktar inte bara objektsinledda och adverbialsinledda satser på detta sätt utan också subjektsinledda. Sålunda analyserar vi "Hunden bet flickan" som: sent(nps(hunden),sdsent(vt(bet),npo(flickan))).

Det betraktelsesätt vi anlagt ovan är högst naturligt för nordiska lingvister, som känner till Paul Diderichsens satsschema där man just avskiljer ett första element, det s.k. fundamentet från

resten. Diderichsen observerar som bekant sedan att de följande konstituenterna alltid kommer i samma ordning och att det finns en tom plats från vilken man kan säga att konstituenten i fundamentet flyttats. Diderichsen är emellertid inte så uppmärksam på de serier av konstituenter som verkligen kan följa på ett visst fundament. Det är dessa serier eller mönster som identifieras av referentgrammatik och kallas subjektsdefekt sats (sdsent), objektsdefekt sats (odsent), adverbdefekt sats (adsent), etc.

Nominalfras och relativsats

Typiska nominalfraser är: "den lille pojken", "flickan vid fönstret", "barnet som lekte". Ur ordklassynpunkt kan vi urskilja den framförställda artikeln "den", adjektivet "lille", det relativa promenet "som" samt gängse verb och andra ordklasser i relativsatsen. En nominalfras kan enligt referentgrammatisk analys bestå av ett huvud (nph) vilket består av ett substantiv föregånget av en eller flera bestämningar inklusive artikel. Substantivet i huvudet styr kongruensen inom nominalfrasen genom att referenten berikas med de grammatiska drag (definithet, genus,numerus) som behövs. Efter substantivet kan förekomma en eller flera prepositionsfraser som kallas "ppa" och dessutom kan det förekomma en eller flera relativsatser "relcl". Förenklat kan vi visa detta genom följande regel som t.ex. kan beskriva: "Barnet på gården som sprang" (de olika upplysningarna A,B,C samlas vid "nom"=namn.

np(R,nom(A,B,C)) --> nph(R,A),ppa(R,B),relcl(R,C).

Regeln visar att samma referentnummer (berikad med grammatiska drag) återfinnes inom nph, ppa och relcl. Vi skall nu se närmare på relativa satser. Som nämnts tidigare är det naturligt att betrakta relativa satser som defekta och relativmarkören som en förmedlare av den saknade konstituenten. Den saknade konstituenten kan vara olika satsdelar och det är då lämpligt att urskilja olika relativa markörer: rels (subjektsrelativmarkör), relo (objektsrelativmarkör), relg (genitivrelativmarkör) och rela (adverbrelativmarkör). Dessa kombineras med motsvarande defekta kategorier: (sdsunt, odsunt, adsunt, odpp, gdnp) såsom exemplifieras av nedanstående analyser.

1. Flickan som sprang

o-representation

np(nph(flickan),relcl(rels(som),sdsunt(vi(sprang))))

funktionell representation

np(1,(flickan,s(subj(1),pred(sprang))))

2. Flickan som hunden bet

o-representation

np(nph(flickan),relcl(relo(som),odsunt(nps(hunden),vt(bet))))

funktionell representation

np(1,(flickan,nom(s(subj(2,hunden),pred(bet),obj(1))))

3. Flickan som hunden sprang till

o-representation

np(nph(flickan),relo(som),odsunt(nps(hunden),vi(sprang),odpp(till))))

funktionell representation

np(1,(flickan,s(subj(2,hunden),pred(sprang),advl(till,1))))

4. Flickan till vilken hunden sprang

o-representation

np(nph(flickan),relcl(rela(p(till),relo(vilken)),adsunt(nps(hunden),
vi(sprang))))

funktionell representation

np(1,(flickan,s(subj(2,hunden),pred(sprang),advl(till,1))))

5. Flickan vars hund sprang

o-representation

np(nph(flickan),relcl(rels(relg(vars),n(hund)),sdsunt(vi(sprang))))

funktionell representation

np(1,(flickan,s(subj(2,(hund,poss(1)),pred(sprang))))

6. Flickan till vars mor hunden sprang

o-representation

np(nph(flickan),relcl(rela(p(till),relo(relg(vars),n(mor)),
adsunt(nps(hunden),vi(sprang))))

funktionell representation

np(1,(flickan,s(subj(2,hunden),pred(sprang),
advl(till,np(3,(mor,poss(1)))))))

I ovanstående uppställning är förkortningen "odpp"= objektsdefekt prepositionsfras, vilken då bara representeras av en preposition. Vi har representerat genitivens motsvarighet i den funktionella representationen med "poss()".

Vi observerar att "som" aldrig kan strykas (vara []) före sdsunt, men väl före odsunt, ett känt faktum. Vi observerar också att "som" inte kan vara objektsrelativ (relo) efter preposition. Man kan inte säga "flickan till som hunden sprang". Det finns ett antal relativa adverbial och konjunktioner som kan analyseras på motsvarande sätt, t.ex. "varest", "varigenom", "då", "när".

Satser med hjälverb eller particip

I referentgrammatik betraktas infinitiver och particip som "minor sentences", satser som saknar en eller flera konstituenter, vilka kan fyllas i från omgivningen. En sats som "Per vill simma" får denna funktionella representation:
s(subj(Per),pred(vill),obj(s(subj(X),pred(simma))))
I denna representation kan man direkt sätta Per som X eftersom det ju är Per som skall simma. Subjektet i sådana satser med kedjor av verb är aktuellt som subjekt även i senare infinitiver som visas av: "Per vill försöka sluta röka". Analysen med infiniven som objekt stöds också av frågor som: "Vad vill Per?" Ord- och frasanalysen av "Per vill simma" är: sent(nps(Per),sdsent(mod(vill),isent(simma))) där "mod" betecknar modalt verb och "isent" betecknar infinitivsats.

En infinitivsats kan sakna en satsdel (förutom subjektet) såsom framgår av en mening som: "Vem vill hunden bita?" där "Vem" måste vara objekt till "bita" i den funktionella representationen: s(subj(hunden),pred(vill),obj(s(subj(hunden),pred(bita),obj(vem)))).
För att hantera detta räknar referentgrammatik med defekta infinitivsatser också, i detta fall en "odisent" (objektdefekt infinitivsats).

På motsvarande sätt behandlas particip. Satsen: "Per har hämtat bilen" har den funktionella representationen: s(subj(Per),pred(har), obj(pfsent(subj(Per),pred(hämtat),obj(bilen)))). I satsen "Bilen är hämtad" måste däremot "bilen" bli objekt till participet "hämta".

Avslutning

Som framgått kan man använda referentgrammatik i menings- och textanalys "för hand", men dess styrka är att den är så systematiskt och konsekvent formulerad att den också kan användas för datoranalys och datorsyntes av meningar. Vi har antytt ovan hur det går till. De fragment av svenska engelska och några andra språk som

programmerats och kan köras på VAX eller PC kan också användas för automatisk översättning, men det krävs ofta transferregler där den funktionella representationen för meningen i ett språk överförs till den motsvarande funktionella representation i det andra språket. Även om de funktionella representationerna är abstrakta och avlägsnar mycket av de enskilda språkens "lokala" idiosynkrasier så är de ofta inte tillräckligt genereralla. Vi skall emellertid inte gå in på dessa problem här.

LITTERATUR

Chomsky,N. Lectures on Government and Binding. Dordrecht (1981:Foris)

Diderichsen, P. Elementaer dansk grammatik. København (1946)

Gazdar,G., E.Klein, G.Pullum & I.Sag, Generalized Phrase Structure Grammar. Oxford (1985: Basil Blackwell)

Sells,P. Lectures on contemporary syntactic theories. CLSI, Stanford(1985)

Sigurd,B. Referent grammar (RG). A generalized phrase structure Grammar with built-in referents. Studia Linguistica 1987:2

Sigurd,B. A referent grammatical analysis of relative clauses. Working Paper Dept of Linguistics and Phonetics, Lund 1988.

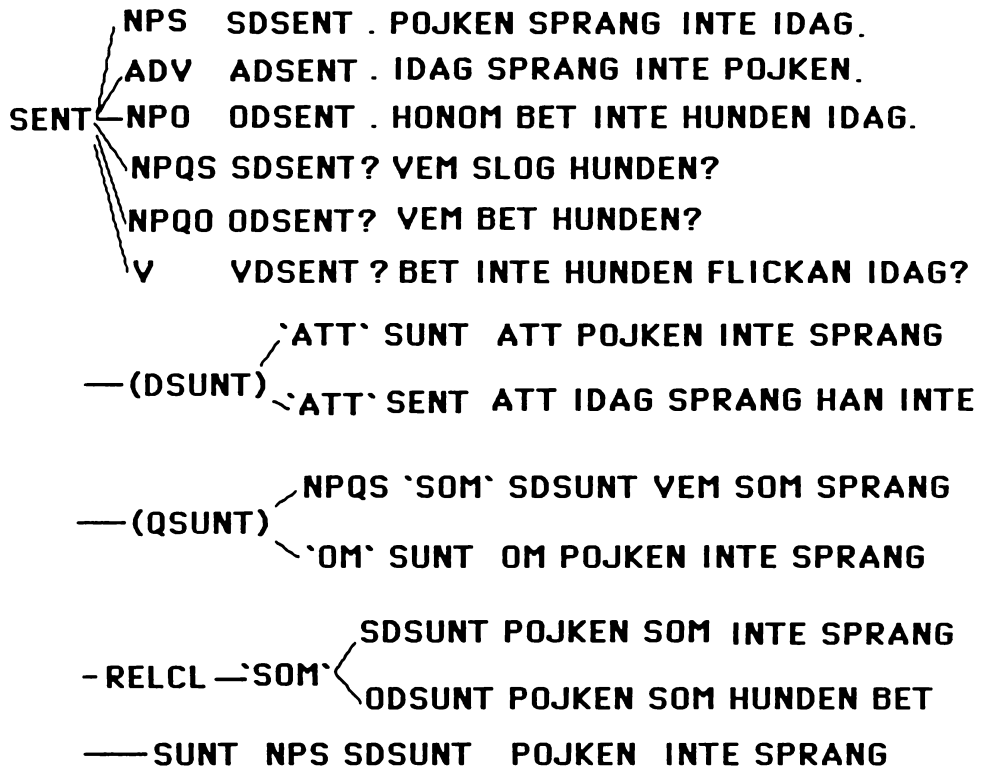
Gawronska-Werngren,B. A referent grammatical analysis of relative clauses in Polish (1988: manuscript)

Föredrag vid Nordiska Datalingvistdagarna i Köpenhamn 3-4 nov,1987

I SWETRA (Swedish Automatic Translation Group, Lund) arbetar också Mats Eeg-Olofsson och Lars Gustafsson (med stöd av HSFR).

APPENDIX (1)

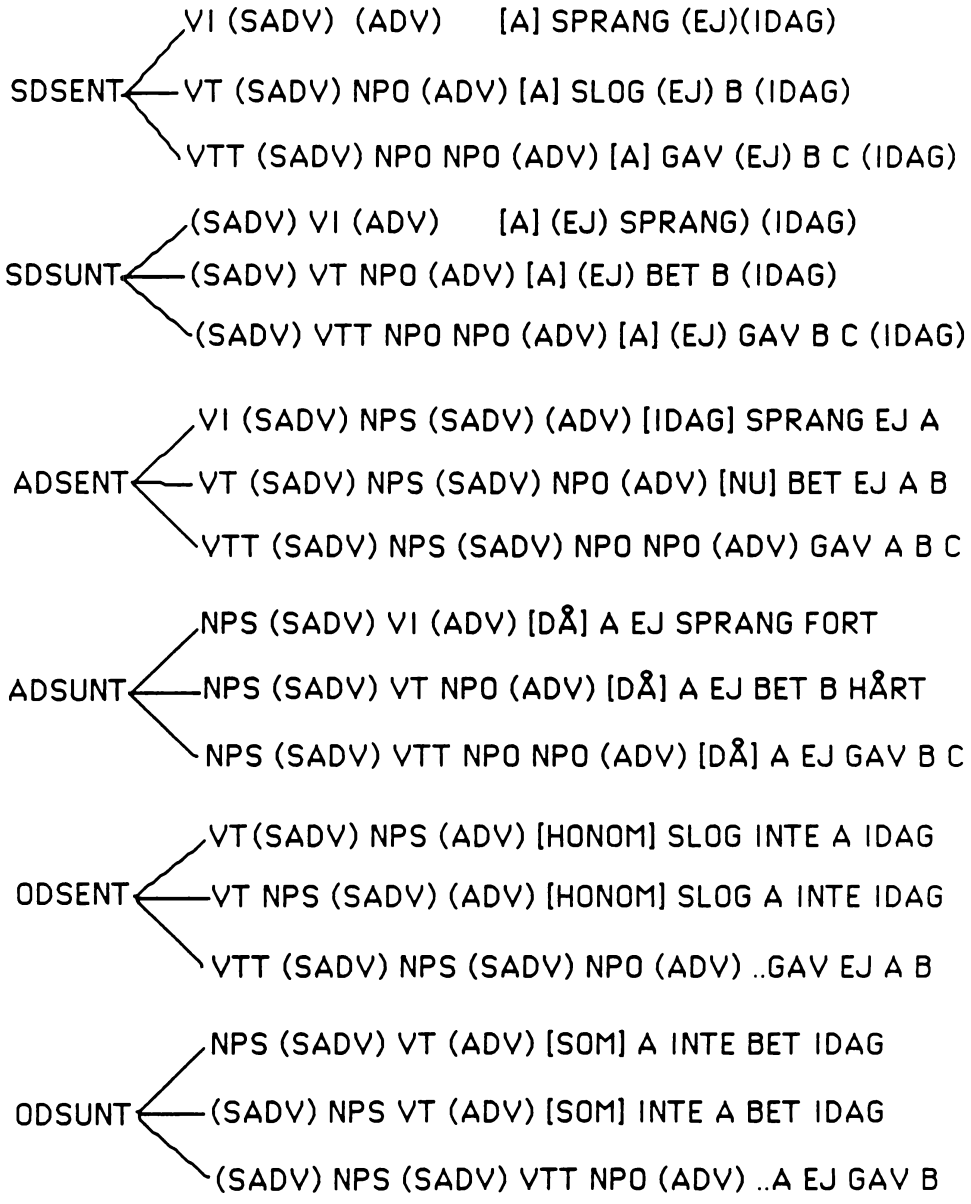
B.SIGURD.REFERENT GRAMMAR (1987)



SOME BASIC TYPES OF MAIN SENTENCES (SENT) AND
SUBORDINATE SENTENCES (SUNT).
SDSENT= SUBJECT DEFECTIVE SENTENCE
ODSENT= OBJECT DEFECTIVE SENTENCE
ADSENT= ADVERBIAL DEFECTIVE SENTENCE
SDSUNT= SUBJECT DEFECTIVE SUBORDINATE SENTENCE
NPS= SUBJECT NP, NPO= OBJECT NP
NPQS= SUBJECT QUESTION WORD (VEM,VAD)
NPQO= OBJECT QUESTION WORD (VEM,VAD)
ADV= ADVERB

NOTE THE DIFFERENT PLACEMENT OF SENTENCE
ADVERBS IN MAIN AND SUBORDINATE SENTENCES

APPENDIX (2) **B.Sigurd,Referent Grammar (1987)**



WORD (CONSTITUENT) ORDER WITHIN DEFECTIVE MAIN (SENT) AND SUBORDINATE (SUNT) DEFECTIVE SENTENCES. SDSENT= LACKING SUBJECT, NPS=SUBJECT NP, NPO=OBJECT NP, VI=INTRANSITIVE VERB, VT=TRANSITIVE VERB, VTT= DOUBLY TRANSITIVE VERB, SADV=SENTENCE ADVERB

APPENDIX (3) Bengt Sigurd, LUND

UTSKRIFTER AV REFERENTGRAMMATISKA DATORANALYSER.

CHECK BEGÄR ANALYS. FÖRSTA RADEN GER MODUS (d,q).

ANDRA RADEN GER FOKUSERAD KONSTITUENT.

TREDJE RADEN GER FUNKTIONELL ANALYS.

```
?- check([pojken,lovade,flickan,att,komma,.]).
d
np(_1, nom(m(boy, sg), m(def)))
s(subj(np(_1, nom(m(boy, sg), m(def)))), pred(m(promise, past)), dobj(np(_2,
        nom(m(girl, sg), m(def)))), aobj(s(subj(np(_2, nom(m(girl, sg),
        m(def)))), pred(m(come)))))
d
np(_1, nom(m(boy, sg), m(def)))
s(subj(np(_1, nom(m(boy, sg), m(def)))), pred(m(promise, past)), dobj(np(_3,
        nom(m(girl, sg), m(def)))), aobj(s(subj(_2), pred(m(come)))))


?- check([pojken,vill,komma,.]).
d
np(_1, nom(m(boy, sg), m(def)))
s(subj(np(_1, nom(m(boy, sg), m(def)))), pred(m(will, pres)), obj(s(subj(np(_1
        , nom(m(boy, sg), m(def)))), pred(m(come)))))
no

?- check([pojken,vill,kunna,komma,.]).
d
np(_1, nom(m(boy, sg), m(def)))
s(subj(np(_1, nom(m(boy, sg), m(def)))), pred(m(will, pres)), obj(s(subj(np(_1
        , nom(m(boy, sg), m(def)))), pred(m(can, inf)), obj(s(subj(np(_1,
        nom(m(boy, sg), m(def)))), pred(m(come)))))))
no



?- check([pojken,har,sprungit,.]).
d
np(_1, nom(m(boy, sg), m(def)))
s(subj(np(_1, nom(m(boy, sg), m(def)))), pred(perf), obj(s(subj(np(_1,
        nom(m(boy, sg), m(def)))), pred(m(run, perf)))))



?- check([barnet,som,pojken,som,hunden,bet,slog,sprang,.]).
d
np(_1, nom(m(child, sg), m(def), s(subj(np(_2, nom(m(boy, sg), m(def),
        s(subj(np(_3, nom(m(dog, sg), m(def)))), pred(v(m(bite, past))),
        obj(_2))))), pred(v(m(hit, past))), obj(_1))))
s(subj(np(_1, nom(m(child, sg), m(def), s(subj(np(_2, nom(m(boy, sg), m(def),
        s(subj(np(_3, nom(m(dog, sg), m(def)))), pred(v(m(bite, past))),
        obj(_2))))), pred(v(m(hit, past))), obj(_1))))), pred(v(m(run,
        past))))
no
```

UTSKRIFTER AV REFERENTGRAMMATISKA DATORANALYSER.

CHECK BEGÄR ANALYS. FÖRSTA RADEN GER MODUS (d,q).

ANDRA RADEN GER FOKUSERAD KONSTITUENT.

TREDJE RADEN GER FUNKTIONELL ANALYS.

```
?- check([vem,sade,pojken,sprang,?]).
q
np(_1, nom(who))
s(subj(np(_2, nom(m(boy, sg), m(def)))), pred(m(say, past)), obj(s(subj(np(_1,
          nom(who))), pred(v(m(run, past))))))


?- check([vem,trodde,pojken,att,flickan,hoppades,att,hunden,bet,?]).
q
np(_1, nom(whom))
s(subj(np(_2, nom(m(boy, sg), m(def)))), pred(m(believe, past)), obj(s(subj(np
         (_3, nom(m(girl, sg), m(def)))), pred(m(hope, past)), obj(s(subj(np(_4
         , nom(m(dog, sg), m(def)))), pred(v(m(bite, past))), obj(np(_1,
          nom(whom)))))))


?-
?- check([honom,sade,pojken,att,flickan,hoppades,att,pojken,sade,att,hunden,bet
.]).
_1
np(_2, nom(m(him)))
s(subj(np(_3, nom(m(boy, sg), m(def)))), pred(m(say, past)), obj(s(subj(np(_4,
          nom(m(girl, sg), m(def)))), pred(m(hope, past)), obj(s(subj(np(_5,
          nom(m(boy, sg), m(def)))), pred(m(say, past)), obj(s(subj(np(_6,
          nom(m(dog, sg), m(def)))), pred(v(m(bite, past))), obj(np(_2,
          nom(m(him)))))))))))
1


?- check([flickan,som,pojken,sade,att,han,slog,sprang,.]).
d
np(_1, nom(m(girl, sg), m(def), s(subj(np(_2, nom(m(boy, sg), m(def)))),
          pred(m(say, past)), obj(s(subj(np(_3, nom(m(he)))), pred(v(m(hit,
          past))), obj(_1))))))
s(subj(np(_1, nom(m(girl, sg), m(def), s(subj(np(_2, nom(m(boy, sg), m(def))))
         , pred(m(say, past)), obj(s(subj(np(_3, nom(m(he)))), pred(v(m(hit,
          past))), obj(_1)))))))), pred(v(m(run, past))))


?- check([vem,trodde,pojken,sade,att,hunden,bet,honom,?]).
q
np(_1, nom(who))
s(subj(np(_2, nom(m(boy, sg), m(def)))), pred(m(believe, past)), obj(s(subj(np
         (_1, nom(who))), pred(m(say, past)), obj(s(subj(np(_3, nom(m(dog,
          sg), m(def)))), pred(v(m(bite, past))), obj(np(_4, nom(m(him)))))))))
```

PARSING DANISH TEXT IN EUROTRA

Ole Togeby
University of Copenhagen
and
EUROTRA DK

Abstract.
The machine translation project Eurotra is described as a multi language modular translation system with 9 monolingual analysis modules, 72 bilingual transfer modules, and 9 monolingual synthesis modules. The analysis module for Danish is described as a 3 step parser with structure generation rules for immediate constituent structure, syntactic structure, and semantic structure, and translation rules between them. The topological grammatical description of Danish proposed by Paul Diderichsen, is shown to be usefull in building the parser for Danish, especially with respect to the interaction between empty slots and filled slot in the topological pattern. At last the special problem with parsing and disambiguation of sentences that allow many pp attachments patterns is mentioned and a solution is suggested.

Introduction

The Council of the European Communities decided in November 1982 to launch a research and development project aimed at the production of a pre-industrial prototype machine translation system of advanced design covering all the official languages in the Community. This project is called Eurotra, and it is a multilingual machine translation system covering 72 language pairs, each of the nine EEC languages being translated into all the other EEC languages. Eurotra is run on a collaborative basis by decentralized groups. In this article I will describe some of the problems we have had in the Danish language group working with translation to and from Danish. So what is reported here is the result partly of the 'linguistic legislation' common for all the language groups i Eurotra, partly of the work in the Danish language group from which many persons have participated in the discussions about how to build a parser of Danish.

The translation is performed in three stages using three independent modules: 1) a source language analysis module consisting of a source language monolingual dictionary and a parsing grammar yielding an interface structure which is language independent formal tree representation of the sentence, decorated with the lexical material from the source language text; 2) a transfer module using a bilingual dictionary by which the lexical items are translated into the target language, and using translation rules by which the interface structure is transferred into, in most cases, an identical target language interface representation; 3) a synthesis module consisting of a monolingual target language dictionary and a grammar, in many respects a mirror image of the grammar used in analysis of that language; this module generates the target language text from the transferred interface representation.

Because the whole translation system consists of 72 transfer modules, but only of 9 analysis modules and 9 synthesis modules, we try to make as much of the work in analysis as possible, yielding an interface respresentation which is the same for the translational equivalents of the source language and target language. The 'only' difference between the interface representations is the lexical material of the sentence being translated.

In this article I will describe the analysis module used by the Danish language group in Eurotra. The parsing of a sentence is done in 3 steps, primarily to provide modularity so that it is easy for all the linguists working in the project to recognize what is

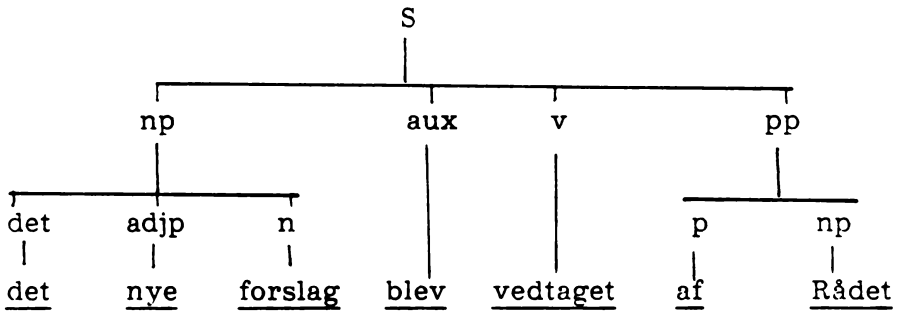
going on in the grammar rules, and so that errors can easily be found and corrected.

From the natural language text we parse to a level called Eurotra Constituent Structure, ECS, where the immediate constituents of the sentence are represented in a tree as np, auxiliary, v, advp and pp, and the immediate constituents of these sentence constituents are represented as daughter nodes with the names adjp, determiner, quantifier, cardinal and so on. From ECS we translate to a level called Eurotra Relational Structure, ERS, where the grammatical constituents of the sentence are represented in a tree with decorated nodes as subject, main verb, object, indirect object, attributive object, complement and modifier, and the constituents of these constituents are represented as modifiers and complements. From ERS we then translate into the Interface Structure, IS, where the dependency structure constituents of the sentence are represented in a tree in canonical order as: first: the predicate, i.e. the verbal head of the sentence, then: argument 1, 2 and 3 of the predicate, and finally sentence modifiers, and the dependents of the dependent constituents as arguments or modifiers of their heads.

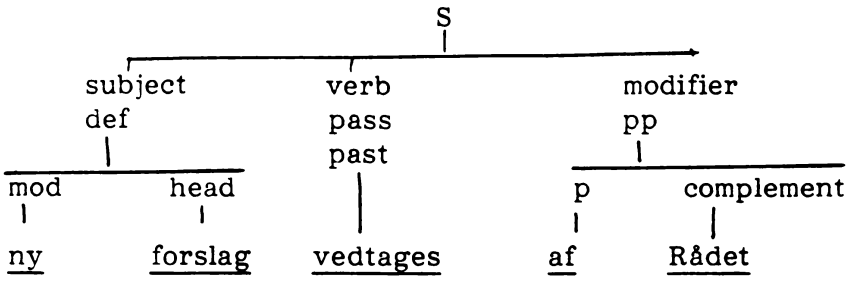
An example can illustrate the parsing process from text to IS:

text: Det nye forslag blev vedtaget af Rådet.

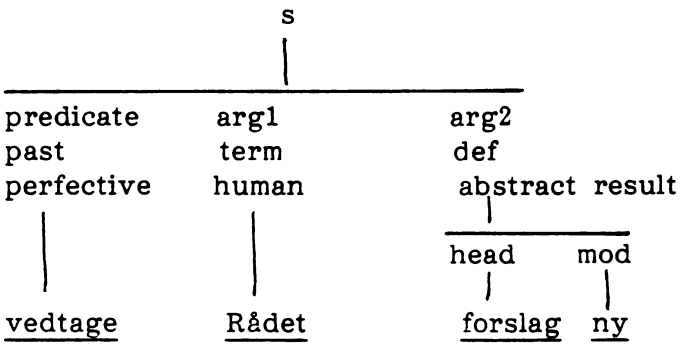
ECS:



ERS:



IS:



This parsing strategy means that we use three types of rules: 1) building rules, which are normal phrase structure rewriting rules. These rules generate the tree structure on each level. 2) Feature rules create the feature decorations on each node of the tree and exclude (kill) generated trees where the features do not match according to the feature match rules specified in the grammar. 3) Translation rules translate a decorated tree from one level into another decorated tree on the next level. In analysis the order of the levels is: text - ECS - ERS - IS, and in synthesis the order is: IS - ERS - ECS - text.

In the next paragraph I will describe some of the problems we have met making an ECS parser of Danish, using Paul Diderichsens topological grammar for Danish.

Overgeneration in a topological parser

It is not surprising that the parsing strategy will not be the same for case languages as Finnish or German and a non free word order language as Danish. A morphological parser has proved to be very efficient for languages with a rich morphology, but it is not at all sufficient for languages where much of the grammatical information is found in the word order. The alternative to a morphological parser is a topological parser, where the information found in the order of the words is transformed into the grammatical tree with canonical order of the decorated nodes.

But it is not clear how to write phrase structure rules generating a grammatical analysis, using the knowledge of the topology of Danish sentences, without overgeneration, i.e. without making many wrong analyses of a given sentence in addition to the wanted analyses.

As described by Paul Diderichsen in Elementær Dansk Grammatik, (Diderichsen, 1946) and elsewhere (Diderichsen, 1945) the order of the constituents in a Danish sentence is the following:

Base	//	actualisation field			//	content field		
	//	v^f	np	$advp^1$	//	v^{if}	np np	$advp^2$
så	//	ville	Petra	ikke	//	følge	børnene	hjem
then		would	Petra	not		follow	the children	home

And in subordinate clauses the order of the constituents is the following:

con-	//	actualisation field			//	content field		
junction	//	np	$advp^1$	v^f	//	v^{if}	np np	$advp^2$
hvis	//	Petra	ikke	ville	//	følge	børnene	hjem
if		Petra	not	would		follow	the children	home

The idea of this topological description is that this pattern is the order of the constituents in the sentence if they are all present in the same sentence; it is a maximally filled frame. If all the slots in the frame are not filled, the internal order of the constituents present in the sentence, will be the same:

Base	// actual. field			// content field				// Heavy
	// v^f	/ np^1	/$advp^1$	// v^{if}	/ np^2	np^3	/$advp^2$	// field
derfor	// har	/ Rådet/		// vedtaget	/ planen/			//
derfor	// vedtog	/ Rådet/		//	/ planen/			//
Rådet	// vedtog	/	/	//		/ planen/		//
i 1982	// sendte	/ Rådet/		// / Kommissionen		forslaget/		//

Literal, i.e. wordorder preserving translation of the sentences:
derfor har Rådet vedtaget planen
therefore has the Council passed the plan

derfor vedtog Rådet planen
therefore passed the Council the plan

Rådet vedtog planen
The Council passed the plan

i 1982 sendte Rådet Kommissionen forslaget
in 1982 sent the Council the Commission the proposal

The positions in this maximally filled scheme correspond systematically to the grammatical functions of the constituents:

In the actualization field the np^1 position after the v^f position is the slot for the subject and the $advp^1$ is the slot for the sentence adverbial; in the content field the np^2 is filled by the indirect object, np^3 by the direct object, and the $advp^2$ position consists of the adverbials modifying the main verb.

In the base all kinds of constituents can be found, except the finite verb; in fact they are moved from their normal position to the base position of the sentence if they are topicalized or marked for contrast to something in the preceding sentence. When a constituent is moved to the base position its grammatical function is indicated by the fact that its position slot in the frame will be empty - a rule which holds for the Germanic languages except for English. In Danish the position of the subject is after the finite verb when something else but the subject is topicalized in the base position; but in English the subject remains in front of the finite verb even if some other constituens, as for example the object, have been topicalized.

In the pedagogical practice where students are taught how to fill

in the words in the slots of the pattern correctly, it is said that if you can not see whether a word in the base is, say, subject or object, you move another constituent in the base position than the one which is there, and then you can see from which slot it has been moved: What is the function of Den plan in the sentence Den plan vedtog Rådet ikke enstemmigt? Put the constituent back again to the position from where it has been moved: Rådet vedtog ikke den plan enstemmigt. Answer: Den plan is the object moved from the content field to the base position.

In addition to the three mentioned fields, there is an final field, called the 'heavy' constituent field, because only heavy np constituents, i.e. constituents consisting of many words, often whole clauses, are placed there for stylistic reasons. The constituent placed in the heavy field is moved from either the np^1 position, the np^2 position or the np^3 position without any change in their grammatical or pragmatical function. But it is only placed there, and you can only see that it is placed there, if the $advp^2$ position is filled, normally with a one word constituent. So the h position is never filled when the $advp^2$ is empty. And if $advp^2$ is filled, the np constituent is either placed in its normal position in actualization field or content field or it is moved to the heavy field:

Base	// actual. field			// content field				// heavy
	// v^f	/ np^1	/ $advp^1$	// v^{if}	/ np^2 np^3		/ $advp^2$	// field
derfor	// har	/ Rådet		// taget	/ forslaget		/ op	//
derfor	// har	/ Rådet		// taget	/		/ op	// det forslag der skulle imødegå alle de mulige invendinger der kunne komme fra 3. landes side
Rådet	// opvervejer/	/		//	/at vedtage planen/			//
Rådet	// tøver	/	/	//	/		med	//at vedtage planen
derfor	// har	/ Rådet/	ikke	// anbefalet/	Kommissionen			at vedtage planen/ //
derfor	// har	/ Rådet/	ikke	// givet/	Kommissionen	tilsagn/om//		at vedtage planen

Literal translation of the Danish sentences:

derfor har Rådet taget forslaget op
therefore has the Council taken the prosal up

derfor har Rådet taget op det forslag der skulle imødegå
therefore has the Council taken up the proposal which should oppo-

alle de mulige invendinger der kunne komme fra 3. lande
se all the possible objections which could come from 3 countries

Rådet overvejer at vedtage planen
The Council considers to pass the plan

Rådet tøver med at vedtage planen
The Council hesitates with to pass the plan

derfor har Rådet ikke anbefalet Kommissionen at vedtage
therefore has the Council not recommended the Commission to pass

planen
the plan

derfor har Rådet ikke givet Kommissionen tilsagn om at
therefore has the Council not given the Commission promise about to

vedtage planen
pass the plan

If you should write formal rewriting rules which can be implemented and run in a computer, this knowledge of the topology of the Danish sentence could be formulated in a formal (ECS) grammar like this:

(^x indicates that the x is optional, i.e. occurs zero or one time,
*x indicates that x occurs zero, one, or more times.)

G.I.
1. S -> ^b, v^f,^np, *$advp^1$, *v^{if}, *np, ^prt, *$advp^2$, ^h

2. b -> np
$advp^2$,

3. h -> v^2, *np, *$advp^2$, ^h

np, *np
sc (subordinate clause)

4. $advp^2$ -> adv^2
pp

5. pp --> p, np

6. np -> ^detp, *adjp, n, *pp, ^sc

7. sc -> ^conj, ^np, *adv^1, v^f, *v^{if}, *np, *$advp^2$.

This grammar will give the correct analysis of most Danish sentences (except for some refinement about 'light' constituents, and a special negation position which I will not discuss here). All positions except the finite verbs are optional; so a given position may be filled by the constituent that fits into the slot, or it may be empty if no constituent fits into the slot. But the problem is that when the analysis of a sentence is computed not only the correct analysis will be the result, but also a lot of wrong analyses.

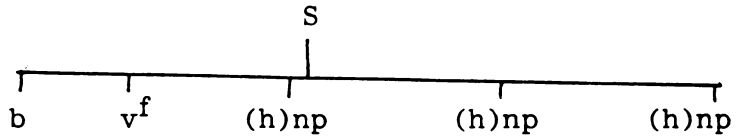
Here it is necessary to distinguish between sentences which from a grammatical point of view are ambiguous, and sentences which are grammatical unambiguous but will nevertheless result in grammatical wrong analyses in addition to the correct one.

If we analyse the sentence Adam elskede Eva, 'Adam loved Eve', we want the machine to give two analyses: one with Adam as subject placed in the base and Eva on np^3, and one with Eva as subject placed on np^1 and Adam as object placed in the base, corresponding to Adam måtte elske Eva and Adam måtte Eva elske respectively. The same will hold for the sentence Dette forslag vedtog Rådet, literal translation: 'this proposal passed the Council'; from a purely grammatical point of view this second sentence is ambiguous in the same way. This problem cannot be solved by a grammatical parser.

The problem with the grammar G. I is that it will give 6 analyses of the sentence: I 1982 sendte Kommissionen Rådet forslaget , literally:'in 1982 sent the Commission the Council the proposal' although it is not grammatical ambiguous:

^b	v^f	^np	*advp^1	*v^{if}	*np	*advp^2	^h
i 1982	sendte	Kom.		Rådet forslaget			
i 1982	sendte			Kom. Rådet forslaget			
i 1982	sendte	Kom.			Rådet		forslaget
i 1982	sendte				Kom.Rådet		forslaget
i 1982	sendte				Kom.		Rådet forslaget
i 1982	sendte						Kom. Råd. forsl.

And in all 6 cases the tree structure will be the same:



In other words the parsing in the machine according to G.I. would yield 6 resulting trees with the only difference that in some of them one, two or three of the last np's would be represented by a mother node h.

The problem is that the interrelation between the empty slots in the pattern is not taken into account by the rules. The interrelations are in this example: np^1 will only be empty when the subject is placed in b; np^2 will only be filled in if np^3 is filled in; h will only be filled by an np if either np^1 or np^{2-3} is empty and $advp^2$ is filled. The hat, ^, indicating optionality, and the star, *, indicating iterativity are not contextsensitive, so the interrelations cannot be reflected in the rules of G.I.

The Danish Eurotra-parser

Because of the overgeneration of the G.I grammar, the linguists in the Danish language group have built a grammar in which we have tried to describe thee interrelation between filled slots and empty slots. It looks like the following:

G.II.

1. s -> (^conj, sva, *v^{if}, ^npp, *$advp^2$, ^sc
(^conj, vsa, *v^{if}, ^npp, *$advp^2$, ^sc

2. sva -> np, v^{f}, $advp^{1}$

3. vsa -> ($advp^{2}$, v^{1}, np, '$nadvp^{1}$
(ap, v^{1}, np, '$nadvp^{1}$
(pp, v^{1}, np, '$nadvp^{1}$
(sc, v^{1}, np, '$nadvp^{1}$
(np (demonstrative), v^{1}, np, '$nadvp^{1}$

ap = adjectival phrase
(= either... or
(

4. sc -> sbb, *v^{if}, ^npp, *$advp^{2}$

5. sbb -> (np, *$advp^{1}$, v^{f}
(subconj, np, *$advp^{1}$, v^{f}
(relpron, np, *$advp^{1}$, v^{f}
(relpron, *$advp^{1}$, v^{f}

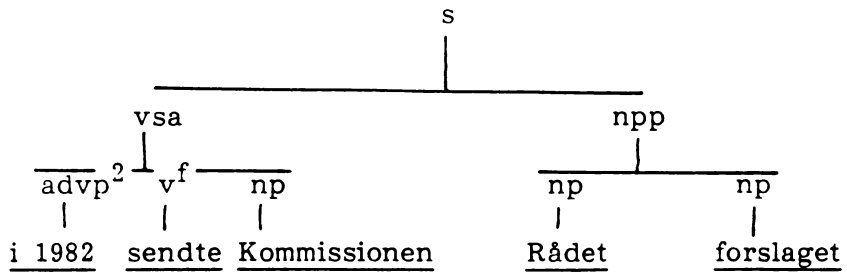
6. npp -> (^np, np
(^np, ap
(^np, sc

7. np -> ^detp, *ap, n, *pp, ^sc

8. $advp^{2}$ -> (prep, ^h
(prt, ^h
(pp, ^h

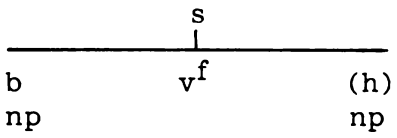
9. h -> (*$advp^{2}$, v^{if}, ^npp, *$advp^{2}$, ^h
(^np, np
(sc.

This G.II. will generate deeper trees than G. I because of the intermediate nodes sva, vsa or npp. But it will only generate one analysis of the sentence: I 1982 sendte Kommissionen Rådet forslaget:



The reason is that np^1 is only filled in if something else but the subject is placed in the base; it means that rule 2. cannot be used; and np^2 will only be filled if np^3 is filled according to rule 6; and h will only be filled if $advp^2$ is filled according to rule 7.

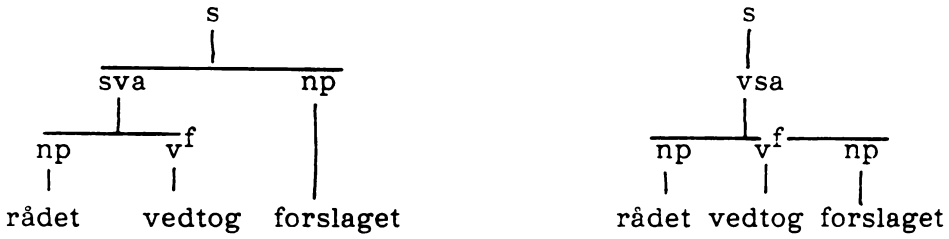
Both G.I and G.II are sets of ECS building rules, but G.II will make the translation rules from ECS to ERS much simpler than G.I would, even in the cases of grammatical ambiguity. Take the example: Rådet vedtog forslaget. G.I will create three nearly identical trees:



And from each of the three created trees the transformation rule used would be:

1. b(np), v^f, np => (subj, vb, obj
   (obj, vb, subj.

G.II would only create two trees out of the sentence:



And there would be one translation rule for each tree:

1. sva(np, v^f), np => subj, vb, obj
2. vsa(np, v^f, np) => obj, vb, subj.

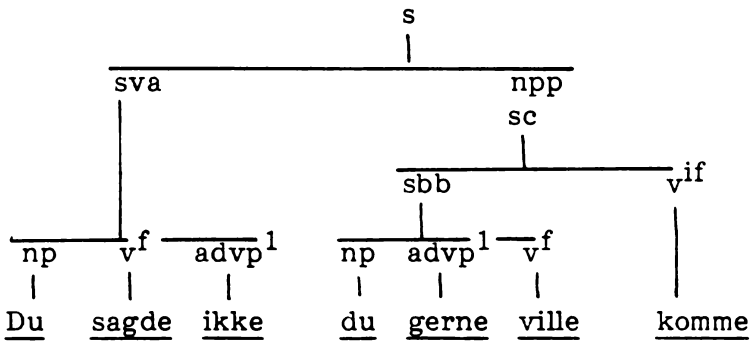
So G.I and the corresponding translation rules would create 6 ERS analyses of the sentence, while G.II and the corresponding translation rules will only create 2 ERS analyses of the sentence.

G. II is better than G.I in disambiguation power because the grammatical information indicated by the word order is used for disambiguation by G.II every time it is present, and the information can be indicated by the fact that a slot is not filled. In the sentence it is indicated that forslaget is not in the heavy constituent field, because adv^2 is not filled.

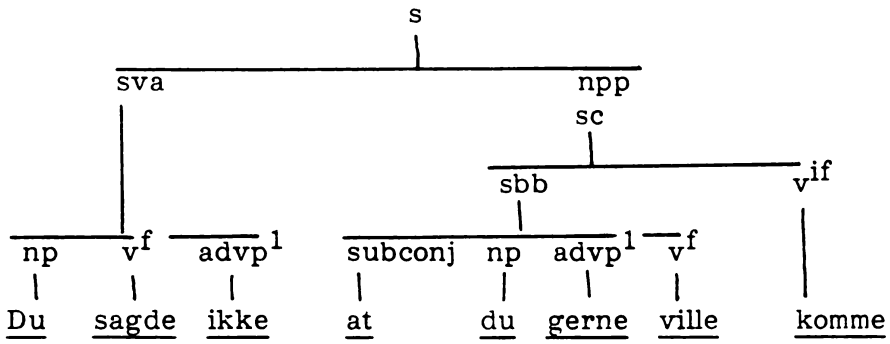
So the generalisations of a topological grammer, the topological interrelationship between constituents, the fact that one constituent can only have a certain position if another constituent has another position, can be registered by a grammar like G.II using more cycles in the generation, i.e. deeper trees with mother nodes indicating the word order of the sentence.

The G.II grammar has been designed by the Danish language group to solve quite a lot of the problematic examples in Danish. In the following I will show some examples of resulting analysis trees:

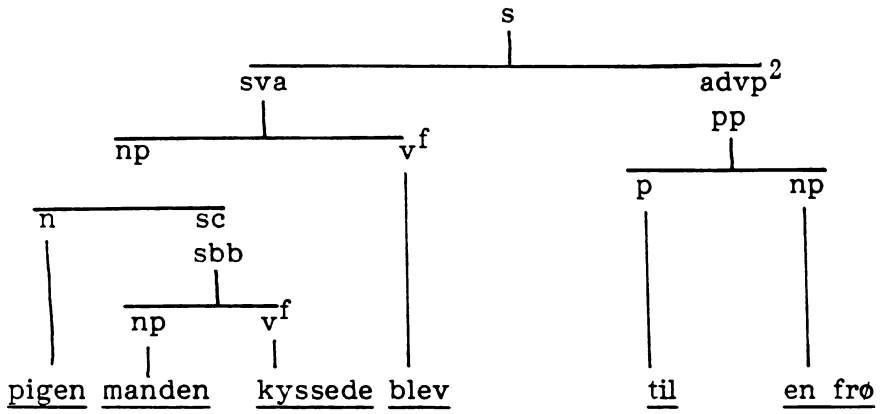
1. Subordinate clauses without conjunction:



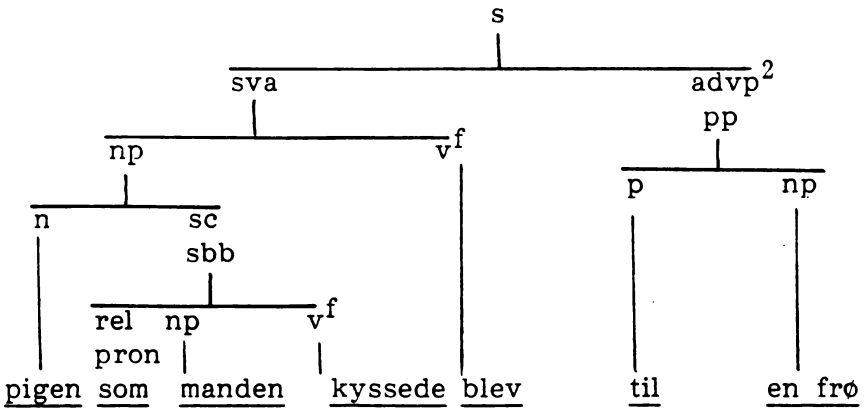
2. Subordinate clause with conjunction:



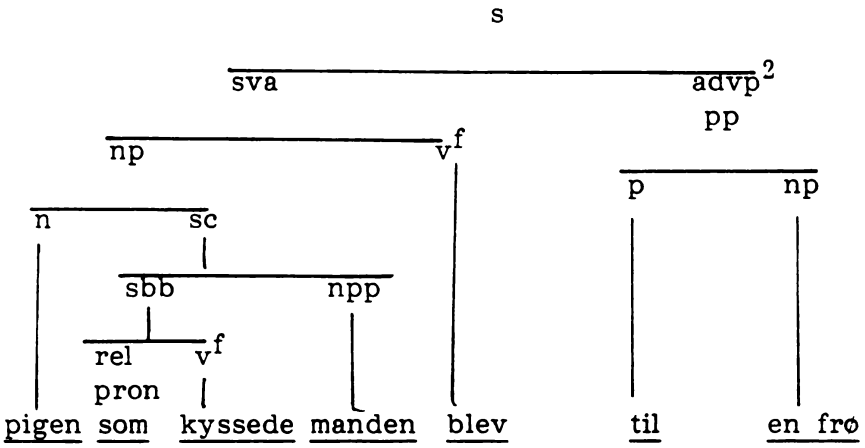
3. Relative clause without relative pronoun:



4. Relative clause with relative pronoun:



5. Relative clause with relative pronoun as subject:



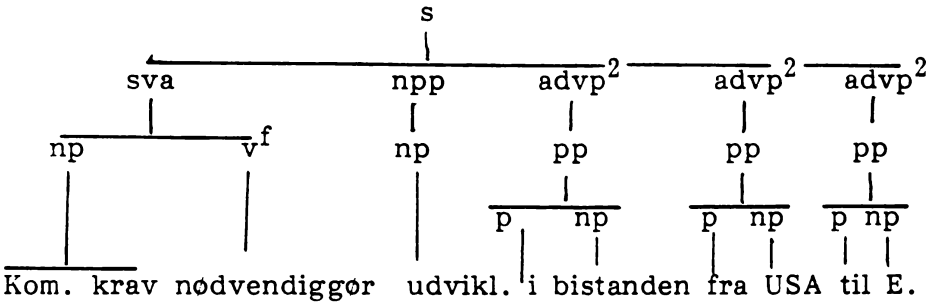
We have not solved all problems in automatic syntactic parsing of Danish sentences: We cannot analyse relative clauses in a 'distance position' , i.e. detached from its head: Europæiske firmaer har taget den udfordring op som ligger i dette emne. The sentence will be parsed by the grammar, but the anaphora from som to udfordring cannot be stated. We cannot parse subordinate clauses with a base: Det betød at hvis aftalen skulle indgås, måtte medlemslandene... And we cannot parse conditional clauses with word order as the main clause: Fortsætter udviklingen ikke, er forudsætningerne bristet.

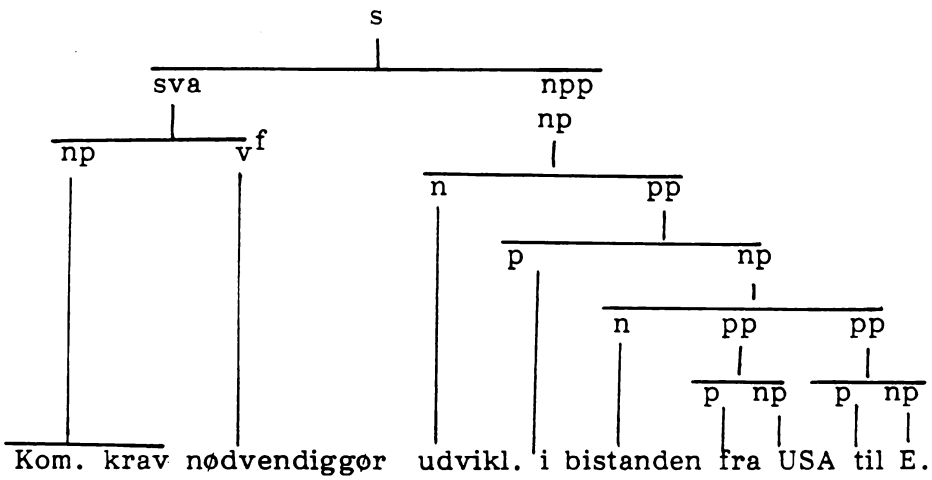
Semantic disambiguation

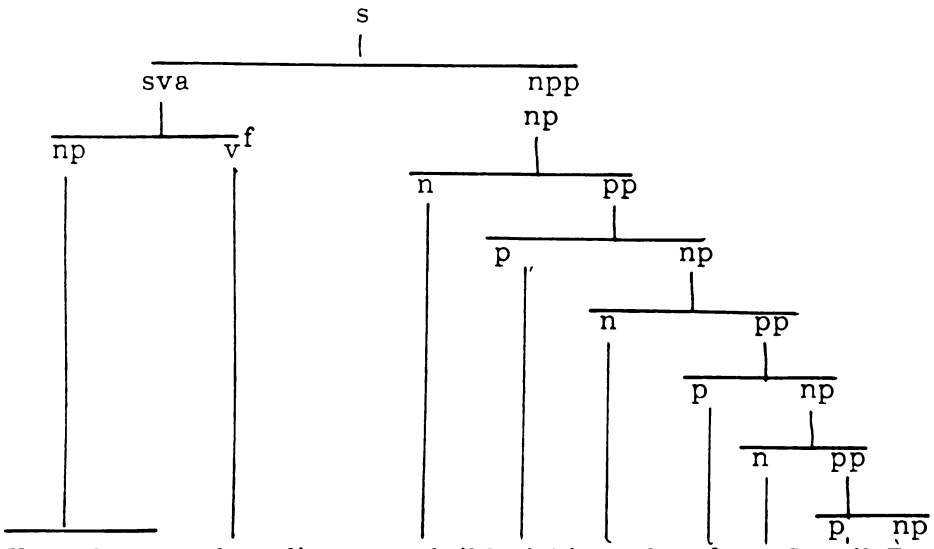
Sentences which are syntactically ambiguous but in many cases semantically unambiguous, are much more frequent than known from traditional grammars. Every time a sentence contains two or more pp's there will be many syntactically acceptable possibilities of pp attachment. The sentence

Kommissionens krav nødvendiggør udvikling i bistanden fra USA til Europa

will have 14 different resulting tree structures, when we parse it with the grammar G.II. I will here give 3 examples of attachment patterns, the flattest tree, the correct tree, and the deepest tree:



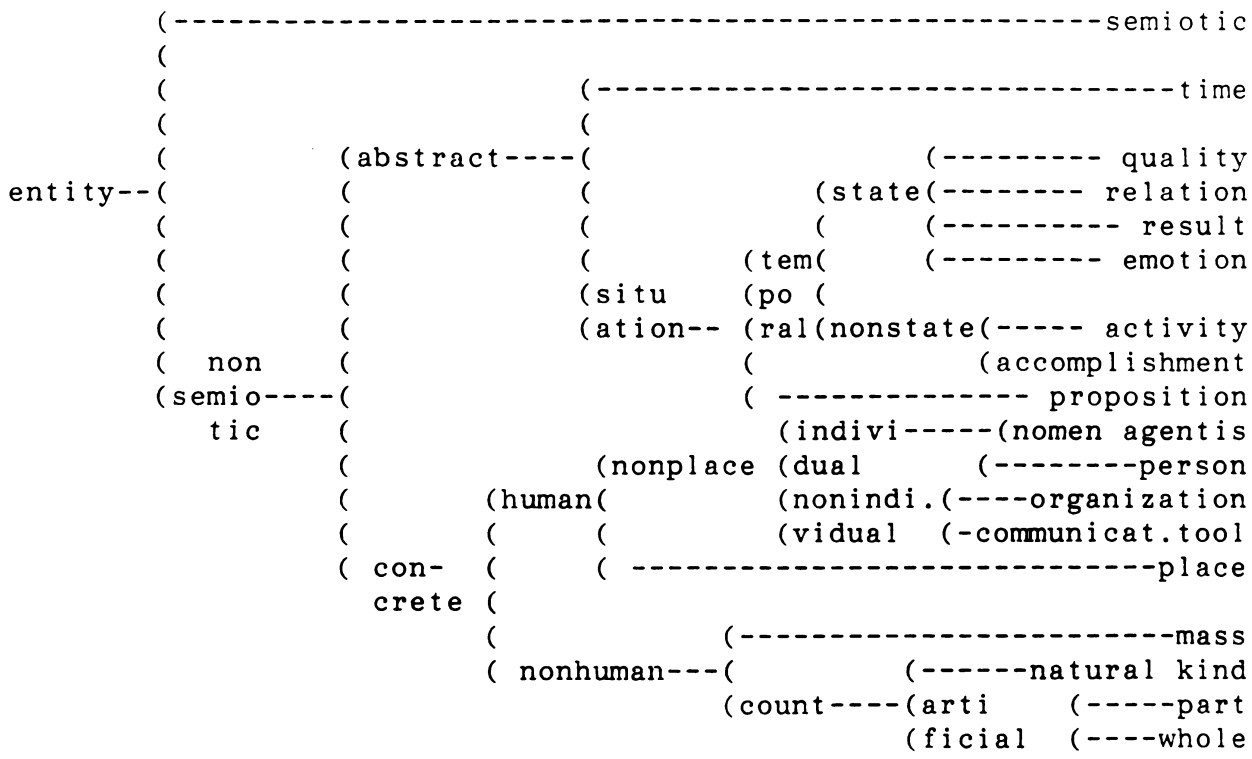




From a purely syntactical point of view all 14 attachment patterns are correct analyses of the sentences, and it is possible to find sentences with each of the 14 structures but other lexical material.

The problem should be solved by use of the feature rules mentioned earlier. What is described in the following is not part of the common Eurotra linguistic legislation, it is not even accepted or discussed in the Danish language group, so the only responsible for the ideas presented in the following is my self.

I imagine that to every noun in the IS dictionary there is assigned a semantic feature with the value chosen among a set og values organised in a hierarchy like the following:



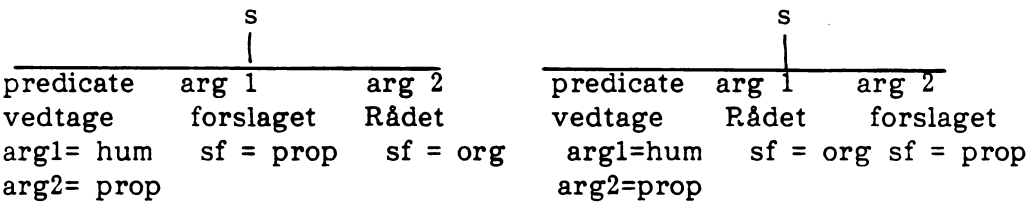
I will not in this paper give the definitions of these features but only show how the system is hierarchically organized, and give a list the lexical entries for the words in the example sentences:

Rådet (semantic feature = organization)
forslag (semantic feature = proposition noun)
Kommissionen (semantic feature = organization)
krav: (semantic feature = proposition noun)
udvikling: (semantic feature = activity)
bistand : (semantic feature = result)
USA: (semantic feature = place)
Europa: (semantic feature = place)

Then to every verb, noun (which has frames), adjective and preposition there is assigned a frame feature specifying the selection restriction from these words to their arguments and modifiers:

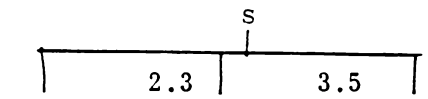
vedtage (sf of argument 1 = human, sf of argument 2 = proposition)
nødvendiggøre: (sf of argument 1 = entity, sf of argument 2 = situation)
krav: (sf of argument 1 = not non human, sf of argument 2 = entity, prep of argument 2 = til)
udvikling: (sf of arg 1 = human, sf of argument 2 = non state, prep of argument 2 = af, i)
i-1: (place where): (argument 1 = place)
i-2: (time during): (argument 1 = time)
i-3: (psychol cause): (argument 1 = emotion)
.
.

bistand: (sf of arg1= hum, sf of arg2 = nonstate, sf og arg 3= hum)
fra-1 (place from where): (argument 1 = not abstract)
.
.
til-1 (place to where): (argument 1 = not abstract)
til-2: (time until): (argument 1 = time)
til-3: ...
.
.
Now for each of the 2 generated is structures of the sentence Rådet vedtog forslaget, and for each of the 14 generated tree structures of the sentence Kommissionens krav nødvendiggør udvikling i bistanden fra USA til Europa, it is computed how well the semantic feature of the argument or modifier matches with the semantic feature selected by the frame of its head. We take the two IS trees :

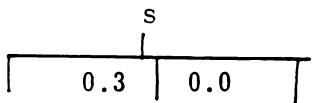


Then we measure the distance in semantic space from the feature value selected by the frame to the feature value of the slot filler in the hierarchy of features by walking from the frame value to the filler value counting 1.0 for every step upwards, and 0.1 for every step downwards. And then the generated tree structure with the

shortest distance from frame value to filler value will be chosen automatically by the machine. This counting is a simulation of how unification works in the program hwen the hiararchy of feature values is implemented. It is possible to implement this preference mechanism.



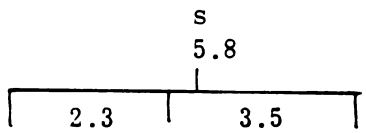
predicate arg 1 arg 2
vedtage forslaget Rådet
arg1= hum sf = prop sf = org
arg2= prop



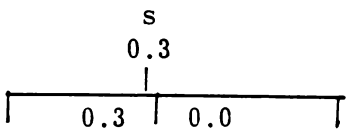
predicate arg 1 arg 2
vedtage Rådet forslaget
arg1=hum sf = orgsf = prop
arg2=prop

So the second tree will be selected by this preference mechanism. It is essential that it is a preference mechanism and not a killer rule which 'kill' all generated trees with mismatch between the value specified in the frame and the value of the slot filler, because if so, all the generated trees, even the wanted one of a metaphorical expression would be excluded: The new framework will solve the problems, the situation threatens to become worse.

If all the 14 generated IS trees of the second example should be computed there is an additional problem: The semantic distances to be compared by the preference mechanism are not distances of unifications in the same node in the tree. So we need to have a adding mechanism so that the two distances measured for argument 1 and argument 2 in the same tree can be added as a total value for the s node:



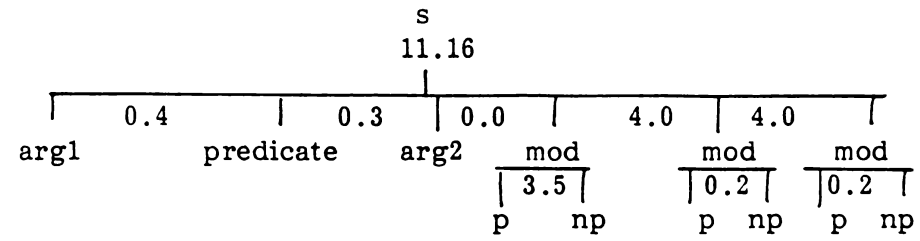
predicate arg 1 arg 2
vedtage forslaget Rådet
arg1= hum sf = prop sf = org
arg2= prop



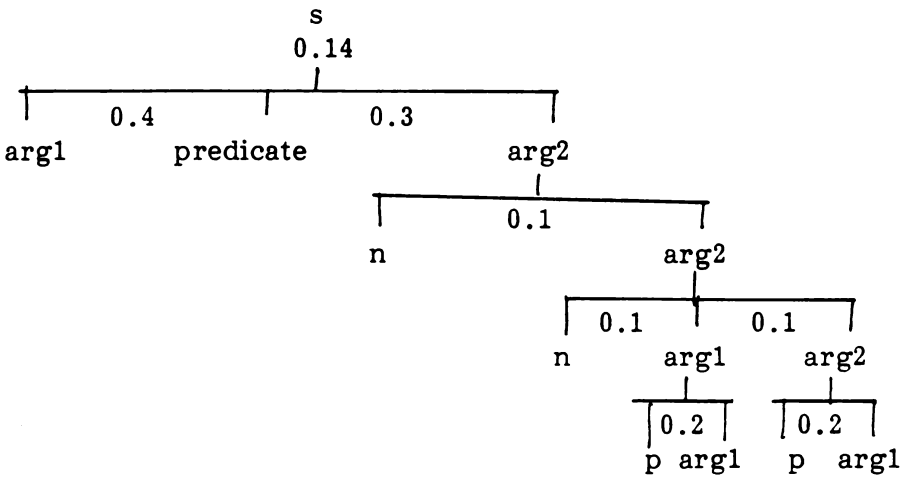
predicate arg 1 arg 2
vedtage Rådet forslaget
arg1=hum sf = org sf = prop
arg2=prop

We have not implemented this mechanism yet. But if it can be done it will turn out that the tree structure which we want is the one which is selected automatically by the preference mechanism in the

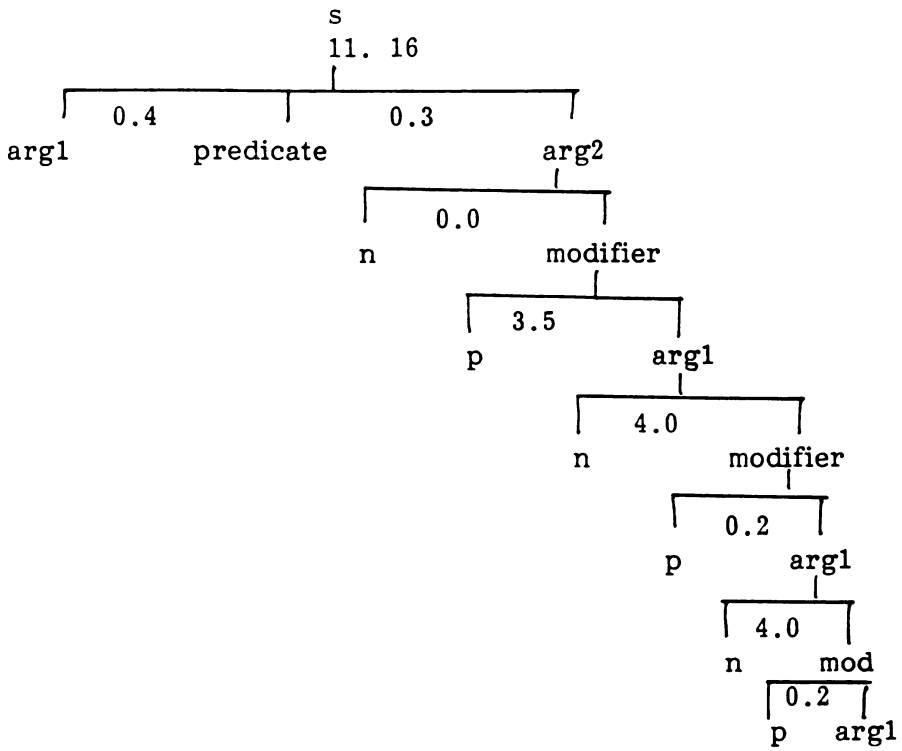
machine.



Kom. krav nødvendiggør udvikl. i bistanden fra USA til E.



Kom. krav nødvendiggør udvikl. i bistanden fra USA til E.



Kom. krav nødvendiggør udvikl. i bistanden fra USA til E.

References:

Diderichsen, Paul 1946. Elementær Dansk Grammatik. Gyldendal, København.
Diderichsen, Paul 1945: Dansk Sætningsanalyse. Dens Formaal og Metode. In
Meddelelser fra Dansklærerforeningen nr. 1 juni 1945.
Reprinted in
Heltoft, Lars og John E. Andersen (eds.) 1986: Sætningsskemaet og dets stilling - 50 år efter. NyS 16-17, Akademisk Forlag, København.

The Eurotra Reference Manual. Version 4.0 December 1987 Luxembourg.

1988: escdk.g Internal Eurotra document containing the Danish ECS grammar

AWARE - DAG-TRANSFORMATIONS FOR SEMANTIC ANALYSIS

Aarno Lehtola and Timo Honkela
KIELIKONE-project, SITRA Foundation
P.O.Box 329, SF-00121 Helsinki
Finland
tel. intl + 358 0 641 877

AWARE is a knowledge representation language for specifying NLU inference rules. AWARE-system takes as its input the parse trees of NL utterances and further refines them by using DAG-transformations (Directed Acyclic Graph) and recursive descent translation techniques. AWARE has been used for semantic analysis in our Finnish language database interface. The input dependency tree is transformed first into a predication DAG and then reduced into a conceptual database query.

Keywords: semantics, graph grammars, inference tools

1 INTRODUCTION

In 1982 SITRA Foundation launched a major project (KIELIKONE) for the study of general computational models for the interpretation of written Finnish. The target application is a Finnish understanding portable database interface.

Currently our hierarchical model of language interpretation consists of six processes: word analysis, lexicalization and disambiguation, sentence parsing, logico-semantic analysis, inference and query adaptation (Jäppinen & al. 1988). All intermediate structures until the so called predication DAG represent more and more refined analysis results (Figure 1). The predication DAG is semantically the richest representation in the model. The following semantic processes simplify and modify the representations towards database queries.

In our model there is a clean separation between linguistic knowledge and processing mechanisms. The extensive use of specialized knowledge description languages characterizes the different components (Lehtola & al. 1987a). There is also a hierarchy of representations. When we work in the morphological stratum, the associated knowledge language deals with sets of features. In the syntactic stratum trees with feature sets in nodes are the dominating representation. In this paper we outline the computational methods used in our logico-semantic stratum which deals with directed acyclic graphs.

DAG-transformations are practical for inferring NL meaning. For instance, they may be used to spesify which NL expressions are near by meaning. Graph rewrite rules may be used to map their syntactic representations into each other. Such rewrite rules would form a meaning-preserving rulebase for a canonizing process. Graph rewrite rules have proved to be convenient also for solving ellipses and anaphoras.

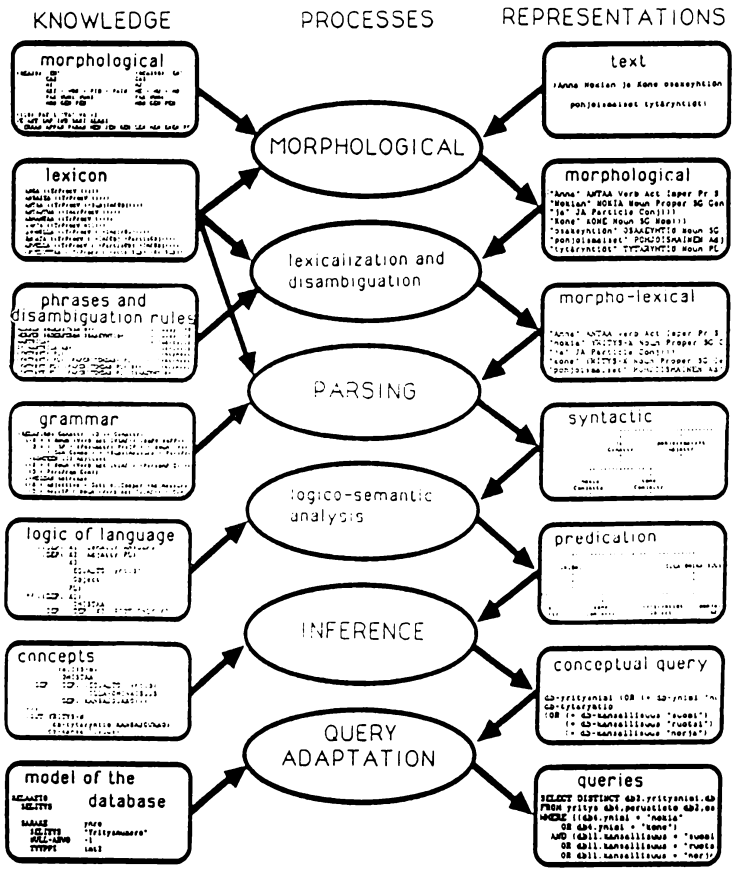


Figure 1. A stratified model of NL interpretation.

In this paper we present the AWARE DAG-transformation system for modelling logico-semantic analysis. First we outline the architecture of AWARE. Then we demonstrate the power of AWARE-rules by examples which deal with canonization of sentence structures, recognition and marking of semantic predications and solution of certain ellipses. Furthermore we discuss how the resulting predication DAGs can be translated into linear expressions, in our case into queries in an universal relation query language. Next we wiev the knowledge acquisition and rule-base maintenance tools. In the end we evaluate the performance of AWARE.

2 THE ARCHITECTURE OF AWARE-SYSTEM

The AWARE-system consists of rule-base maintenance tools and a run-time system (Figure 2). The rule-base is divided into rule packets, which contain rules of

equal priority. Momentarily one or more packets are active. The activation order of packets is specified by a special control language. Each packet has a name and a type. Possible types are 'bottom-up-recursive-scan', 'top-down-recursive-scan', 'wait' and 'transfer'. The type label defines the way the search is carried out.

In AWARE-system the DAGs are usually formed from trees by introducing extra connections. They have one node as the ancestor of all the other nodes. This node we call the root node although in general graph terminology that term is not used. Those nodes which have no descendants we call leaves. The edges are directed out from root nodes towards leaves. The label 'bottom-up-recursive-scan' makes the system to start the search for possible transformations from the leaves and to proceed towards the root node. The recursion comes from the fact that after a succesful transformation the system restarts using the new structure. 'top-down-recursive-scan' works similarly but starts from the root. It is convenient sometimes to let a transformation rule evaluate partially. This is the case when we try to model distant dependencies eg. certain ellipses and anaphora. The rules in 'wait' packets are then used to finnish the incomplete transformations. The label 'transfer' marks those packets which contain attributed rules for recursive descent compilation. These rules are called transfer-rules.

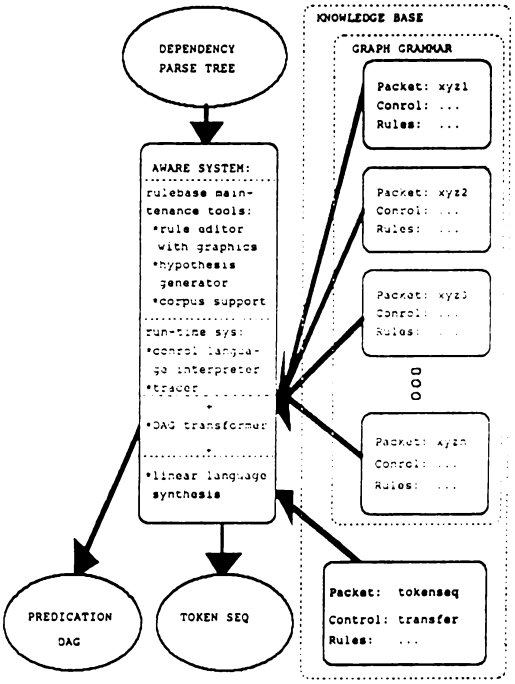


Figure 2. The architecture of the AWARE-system.

The rulebase maintenance tools include a rule editor, a rule hypothesis generator and an automatic book keeping system for corpora. The run-time system contains a control language interpreter, a rule tracer, a precompiler and an interpreter for the actual DAG-transformation rules and for the recursive descent compilation rules.

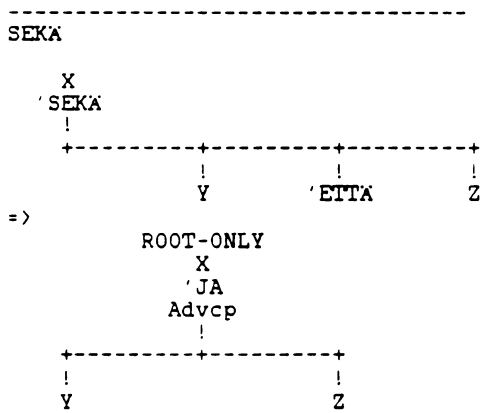
AWARE-rules may function locally without paying attention into larger contexts of the processed constructs. They may also cover whole utterances and on global grounds recognize semantic predications which have their parts syntactically distributed. Rules may amplify themselves by referring to other rules. Also recursive transformations are possible.

An AWARE-description inherits its type definitions from the formal grammar description of the preceding dependency parser (see Lehtola & al. 1985 and Valkonen & al. 1987). A user may also define extra types to be used only in transformations and thereafter (Lehtola & al. 1987b). AWARE is aware of all information that has been derived by the preceding morphological analysis, the lexicalization and disambiguation process and the dependency parser.

3 HOW TO USE TRANSFORMATION RULES

In the following examples we demonstrate the use of transformation rules for different semantic recognition tasks. In many cases the graph transformation reduces into a tree transformation. The first example is very simple, later on we will present more complicated ones.

Canonization of sentence structure



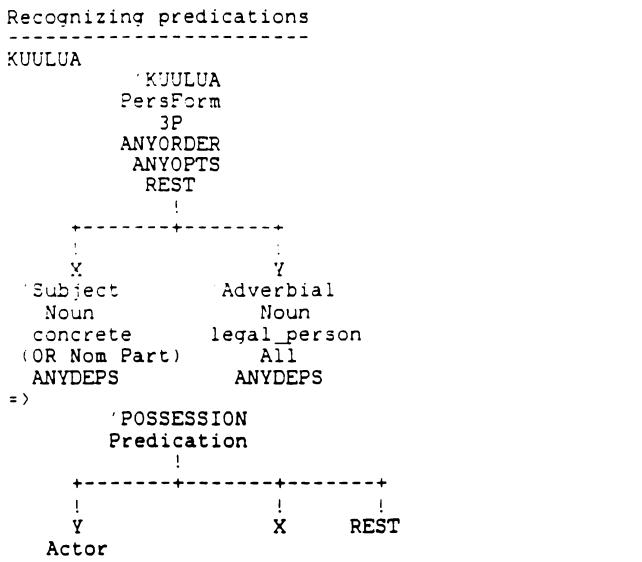
In the rule above one defines a trivial situation where the conjunction phrase SEKA-ETTA (ie. BOTH-AND) is turned to an AND phrase. The rule is composed of a name and two patterns combined with the rewrite operator '=>'. The first pattern

(ie. left-hand-side of the rule, later abbreviated by lhs) shows the topological and feature conditions which will make the rule to perform a transformation.

This rule will be fired if it recognizes a node with the lexeme SEKÄ and with three subordinates on the right. The second subordinate must have the lexeme ETTÄ. X and Y are called glue variables and they are used in the construction of a new structure. The right hand side of the rule forms a new tree to substitute the matched one. When the lhs of the previous rule is matched the glue variable X will have as its value the structure headed by SEKÄ. The glue variable Y is bound to the first subordinate and the variable Z to the third subordinate. The root node of the new structure is the previous SEKÄ node (ie. root only of the structure bound to X) with its lexeme changed to JA and with a role label Advcp. The root will have two subordinates. The first is the same as the first subordinate in the matched tree and the second is the same as the third subordinate in the matched tree.

In the previous rule we provided the lhs-nodes with restrictive feature conditions (eg. 'SEKÄ and 'ETTÄ) and glue variables. In addition it is possible to provide them with the following directives: ANYNUMBER, ANYORDER, ANYOPTS, ANYDEPS. ANYNUMBER states that a node may have unrestrictedly many subordinates of the specified type. ANYORDER lets the subordinates to be located in any mutual order. ANYOPTS states that a node may have unlimited number of optional subordinates. ANYDEPS is a 'wildcard' for totally relaxing the subordinating structures. Finally the nodes may have references to other rules. By inserting a name of a rule into a node one amplifies his definition. In order to satisfy such rule the substructure starting from the marked node must satisfy the named rule.

The rhs-nodes may be provided with features to be over-written (eg. 'JA and Advcp) and with references to glue variables. The directive ROOT-ONLY is used to cut out the connections to the subordinates. In the example rule it is used to cut out the previous connections of 'SEKÄ node (referred by X) so that the new connections introduced in the rhs of the rule would not be overlapping.

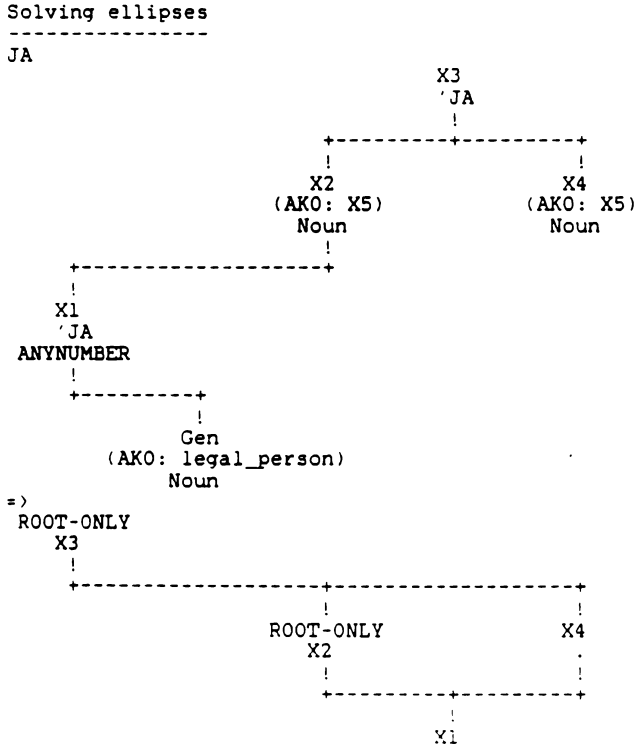


The preceding rule demonstrates how transformations can be used to define word valencies and to detect semantic predications. The example word is the Finnish verb KUULUA (to belong, to be part of, to be audible etc.). The rule describes one instance of the use of verb KUULUA. Here it is stated that the verb KUULUA expresses possession, when it is in a third person personal form and when it has the following two subordinates:

(1) a subject, which is a noun in nominative or partitive case and means a concrete thing

(2) an adverbial, which is a noun in allative case and means a legal person

Both of the subordinates may have any subordinates. The root verb KUULUA may have those subordinates in any order and it may also have unrestricted number of optional subordinates that are to be bound to the set variable REST.



The rule above is an example of how ellipses inside sentences can be solved by DAG-transformations. The rule is semantically restricted to the case of the form "the entity1 and entity2 of leganpersonA, legalpersonB, ... ,and legalpersonX". The dependency tree given by the parser has as its root the conjuntion phrase containing the entities. The two (AKO: X5) expressions test that the nouns X2 and X4 coordinate semantically. The conjunction phrase made of the legal persons is syntactically subordinated only to the first entity. The parser does not recognize the ellipsis that also the second entity is in relation with the same legal persons, The meaning of the rule is that when the described situation is recognized the structure X1 is made to be shared by both of the entities.

Wait-rules for distant bindings

One may leave the filling part (rhs) of a rule partially unspecified. Part of a tree structure is replaced with a call of a wait-rule. Wait-rules are activated afterwards and they look for the matching element from the whole tree structure. Here we demonstrate the use of wait-rules in case of ellipsis. Lets consider the following sentences:

(1) "Anna yritykset, joiden liikevaihto on suurempi kuin metsäalalla keskimäärin!" (Give the companies the turnover of which is greater than the averagenal in forestry)
(2) "Anna yritykset, joiden liikevoiton suhde liikevaihtoon on suurempi kuin metsäalalla keskimäärin!" (Give the companies the ratio of profit and turnover of which is greater than the averagenal in forestry)

Both sentences have an elliptical expression 'the averagenal (turnover/ratio ..) in forestry'. The system cannot locally decide what is the property referred to. By applying wait-rules the decision can be delayed and the larger context is taken into account.

The following rule matches with the expression 'metsäalalla keskimäärin' (the

averagenal in forestry). The rhs of the rule contains a call of a wait-rule. Rule(s) WAIT-PROPERTY specifies different ways of expressing a property of something. See also the label 'AVE' for average which will be attach to the property found.

```
FIELD
              X3
          (AKO: ala)
         'AdjPostAttr
            Common
              SG
              Ad
               !
               +-------+
                       !
                'KESKIMÄÄRIN
                 'Adverbial
     =>
                        'HAVE-PROPERTY
                                  !
               +------------------+--------+
               !                           !
      'HAVE-PROPERTY                WAIT-PROPERTY
               !                        AVE
               !
     +---------+--------+
     !                  !
  'YRITYS           ROOT-ONLY
    PL                 X3
```

The following WAIT-rules match with our examples (1) and (2).

```
WAIT-PROPERTY

              X1
       (AKO: property)

       =>     X1
```

```
WAIT-PROPERTY

                  'SUHDE
                    X3
                     !
       +-------------------------+
       !                         !
(AKO: property)          (AKO: property)

       =>           X3
```

WAIT-rules are contained in a packet of their own. This packet is activated after the packet of the calling transformations has been analyzed. Each call of WAIT-rules may cause only one WAIT-rule to fire. If there is a firing WAIT-rule for each call, the whole dependency structure has been satisfied.

4 RECURSIVE DESCENT TRANSLATION

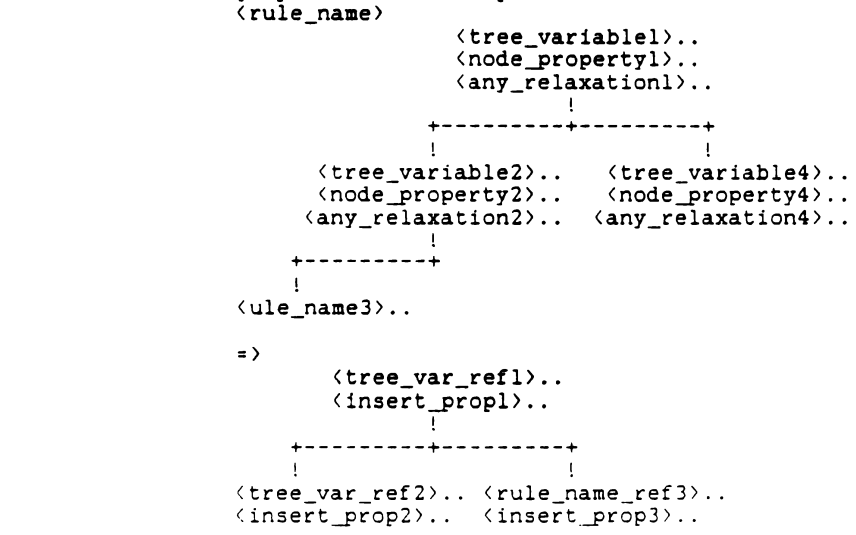
For the production of linear expressions there is an attribute grammar facility. Special translation rules specify the way how different DAG constructs are translated into linear expressions and how a collection of such expressions is mapped into a larger one. The idea is that the transfer rules are seen analogously to the cfg-rules in Knuth's attribute grammars (Knuth 1968). In this

case we are not recognizing a linear language, but rather a graph. The names of AWARE rules correspond to the nonterminals of lhs-parts of cfg-rules. The amplifying rule name references in the match patterns of AWARE rules correspond to the nonterminals in rhs-parts of cfg-rules. In attribute grammar there may be attribute values associated to nonterminals and passed up and down in the constituent tree. In AWARE similar bidirectional attribute value propagation is possible between rules which together cover the recognized graph. In Knuth's grammar attributes are properties of nonterminals, in AWARE they are properties of rule references.

The attribute values may be referred in the equations associated to transfer-rules. These equations consist of tests, assignments, semantic functions and structure building functions. Once a graph is succesfully covered with match-patterns of transfer-rules the equations are instantiated and solved. If there exists solutions the active transfer rules are satisfied, otherwise the search proceeds. The solving process brings the wanted token sequences.

5 RULEBASE MAINTENANCE TOOLS

The rule editor lets the user to manipulate his rules in graphical form. There are two user modifiable representations of the rules, a graphic one and a list expression. The following abstract example demonstrates the parallel use of graphic and list representations:



Is the same as:

```
(⟨rule_name⟩
((DEP: (DEP: ⟨rule_name3⟩)
       ⟨tree_variable2⟩..
       ⟨node_property2⟩..
       ⟨any_relaxation2⟩..)
 ⟨tree_variable1⟩..
 ⟨node_property1⟩..
 ⟨any_relaxation1⟩..
 (DEP: ⟨tree_variable4⟩.. ⟨node_property4⟩.. ⟨any_relaxation4⟩..))
=⟩
((DEP: ⟨tree_var_ref2⟩.. ⟨insert_prop2⟩..)
 ⟨tree_var_ref1⟩..
 ⟨insert_prop1⟩..
 (DEP: ⟨rule_name_ref3⟩.. ⟨insert_prop3⟩..)))
```

The list representations are isomorphic with the graphic representations and the user may choose which ones to edit. The graphical editor supports insertion, modification and deletion of rules. The strength of it becomes apparent when one is doing topological changes into rules. Also nodes are easy to manipulate in graphical form.

The rule hypothesis generator is integrated with the graphical editor. The idea is that by menus the user chooses one of the already created intermediate results to be the lhs of his new rule. The hypothesis generator generalizes the match pattern according to certain heuristic rules and automatically forms a rhs pattern. This rhs pattern is a reduced version of the lhs pattern and the knowledge engineer reforms it by the rule editor. The newly created rule is precompiled and ready for use as the user exists the hypothesis generator. The knowledge acquisition has been very fast by using these tools.

There is an automatic book-keeping facility that records the input sentences and their analysis results into a corpus file. This recording may be done automatically for all input or it may be invoked by the user. The idea is to collect test material to ensure monotonic improvement of knowledge descriptions. After a non trivial change is done in the rulebase, the system runs all test sentences and the results are automatically compared to the previous ones.

6 PERFORMANCE

The AWARE-system has proved to be practical in logico-semantic analysis of Finnish and in query synthesis. It is in daily use in our database interface prototype for a Finnish business database. Total processing of a one line long question takes between 5 and 50 seconds of CPU-time on VAX-11/780. The DAG transformations and the conceptual query synthesis consume about 50 percent of this. The size of the rule base is currently almost 400 rules.

At the first glance the figures may depress. Taken into account that the complexity of the transformational analysis is very high the time consumption is not surprising. At the moment the rules are precompiled into effective data structures, an inverted index is created out of the match conditions and structure sharing is used to minimize memory consumption. Internal data structures are only partly dynamic for the reason of fast information fetch. In spite of the

preceding measures there are still many ways to improve the performance. The current implementation is in FranzLisp.

7 CONCLUSIONS

Compared to certain well known transformation systems (eg. Periphrase of ALPS, MITRE) the AWARE-system offers the following extra properties:

(1) rich type system,
(2) processing generalized for directed acyclic graphs,
(3) orientation towards dependency structures,
(4) powerful tools for knowledge base maintenance,
(5) extensive use of graphics to illustrate the operation,
(6) attribute grammar facility for translation
(7) separate control language
(8) lazy evaluation possible using 'wait' rules

One of the design objectives in AWARE has been to make it so general that it could be used also in machine translation. Dependency structures have been found a good syntactic representation for machine translation purposes. Our dependency parser (Lehtola & al. 1985 and Valkonen & al. 1987) together with AWARE gives interesting prospects for MT.

References

ALPS (1986):
Periphrase Introduction. Report of A.L.P. Systems, Provo, 25 p.

Hobbs, J., Grishman, R. (1976):
The Automatic Transformational Analysis of English Sentences: An Implementation. Intern. J. Computer Math. 1976, Section A, Vol. 5, pp. 267-283.

Jäppinen, H., Honkela, T., Lehtola, A. and Valkonen, K. (1988):
Hierarchical Multilevel Processing Model for Natural Language Database Interface. Proceedings of the 4th IEEE Conference on Artificial Intelligence Applications, San Diego, California, 6 p. (in print).

Knuth, D. E. (1968):
Semantics of Context-Free Languages. Mathematical Systems Theory, vol. 2, no. 2, Springer-Verlag, New York, pp. 127-145.

Lehtola, A., Jäppinen, H. and Nelimarkka, E. (1985):
Language-based Environment for Natural Language Parsing. Proceedings of the 2nd European Conference of ACL, Geneve, pp. 98-106.

Lehtola, A. and Valkonen, K. (1987a):
Knowledge Representation Formalisms and Metadescriptions for the Interpretation of Finnish. Proceedings of the Third Finnish Symposium on Theoretical Computer Science, pp. 64-87.

Lehtola, A. and Honkela, T. (1987b):
AWARE - A DAG Production System with Attribute Grammar Facility *revised report*. Publications of the Kielikone-project, Series B, report 4, Helsinki, 36 p.

Valkonen, K., Jäppinen, H. and Lehtola, A. (1987):
Blackboard-based Dependency Parsing. 10th International Joint Conference of Artificial Intelligence, Milano, pp. 700-702.

Winograd, T. (1983):
Language as a Cognitive Process. Volume I: Syntax. Addison-Wesley Publishing Company, Reading, 640 p.

Zwicky, A., Friedman, J., Hall, B., Walker, D. (1965):
The MITRE Syntactic Analysis Procedures for Transformational Grammars. Proceedings of the Fall Joint Computer Conference 1965, pp. 317-326.

PREDICATION GRAPHS AS CANONICAL REPRESENTATION OF QUERY SENTENCES

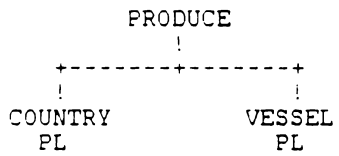
Honkela, T., Lehtola, A. and Valkonen, K.
KIELIKONE-project, SITRA Foundation
P.O.Box 329, SF-00121 Helsinki
Finland
tel. intl + 358 0 641 877

This paper surveys problems encountered in studying the logico-semantic form and discourse problems of Finnish query sentences. We call the logico-semantic form a predication graph. The basic framework we use to represent the logical form of Finnish query sentences is an annotated logical tree transformed from the dependency parse tree using graph transformations of the AWARE-system. Examples of analysing elliptic and anaphoric expressions are given. Finally, some critical points of computational semantics are discussed.

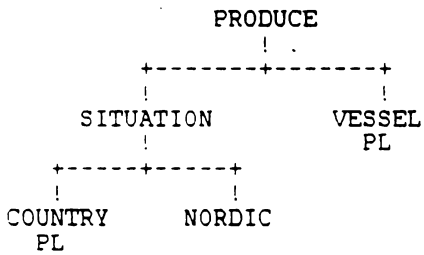
Keywords: semantics, predication, discourse analysis

1 CANONICAL STRUCTURE

One of the main objectives of the logico-semantic level is to give those utterances that differ only syntactically a uniform representation. This representation is a predication structure with predicates and their arguments. For example, the expressions "valtiot, jotka tuottavat laivoja" (countries that produce vessels) and "laivoja tuottavat valtiot" (countries producing vessels) would lead into a single logico-semantic form something like as follows:



The arguments are kind of typed variables. The expression could also be stated: produce(X,Y) and country(X) and vessel(Y). An argument may further be another predication. If the word 'countries' were replaced by 'nordic countries' then the resulting structure would be:



This could be also expressed with a clause: produce(X,Y) and country(X) and situation(X,nordic) and vessel(Y). Our representational form makes the variables in the logical form implicit thus making it more readable. The choise of the variable to be passed to an upper predication is demand driven. The solution is based on the ideas of polymorphism.

2 CHOICE OF PREDICATES

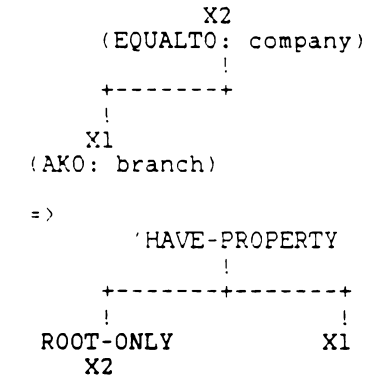
There are two essential decisions to be made when the predicates for logico-semantic form are selected. Firstly, does the system interpret the semantic content of utterances strongly thus making possibly also strong reduction or does the system rely on the original form by using the verbs with their valences as predicates with arguments? Secondly, one must decide whether the predicates are general, specific or both.

2.1 GENERAL PREDICATIONS IN NLI FOR DATABASES

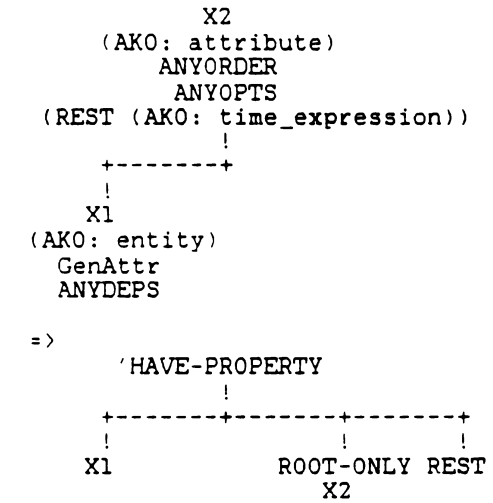
The selection of predications is not determined by the features of the AWARE-system. General predicates as well as specific ones may be used. The degree of canonization depends on the person(s) who makes the semantic modeling, too. In the natural language interface for databases, SUOMEX (Jäppinen & al. 1988) we use general predicates. At conceptual level these predicates reflect the entity-attribute-relationship (EAR) approach for conceptual modeling. This is motivated by the fact that these predicates reflect the conceptual models of the databases.

Let's assume that we have companies with a branch of business and certain properties (or attributes). Here we have an example of using AWARE transformation rules to create predication structure for expressions like "Anna metsäalan yhtiöiden liikevaihto!" (Give the turnover of the companies in forestry).

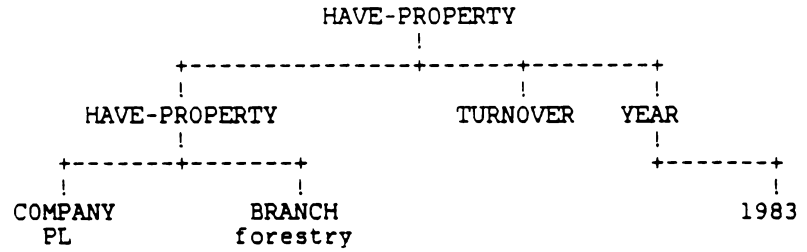
The first rule instance matches to dependency structures (for more about dependency parsing see Lehtola et al.,1985 or Valkonen et al.,1987) where the dependent is restricted to be a particular branch and the regent is any synonym for company. This very simple rule contains only semantic conditions in addition to the dependency structure specified. Further syntactic checks could be added to avoid overgeneration.



The second example is an instance of a rule for covering expressions stating an entity to have a certain property. The entity here could be a company, companies, companies in certain branch etc. The expression is allowed to have an specification of point or interval of time.



The predication structure of the expression "Näytä metsäalan yritysten liikevaihto vuodelta 1983!" (Show the turnover of the companies in forestry in 1983) is shown below.



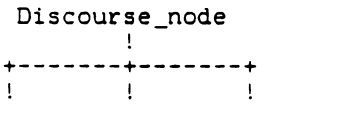
To emphasize the generality of AWARE, one must state that the choice of predicates such as 'HAVE-PROPERTY' follows from its use as a part of a database interface.

2.2 CONCEPTUAL HIERARCHY

Nodes in graph transformation rules may contain semantic restrictions. For each restriction a proper level of generality is needed. Information about conceptual classes is in a form of hierarchy (compare to Grozs et al, 1987). The use of semantic restrictions and their relation to the conceptual hierarchy could be exemplified with pair of expressions like (1) "Peter's car" versus (2) "Peter's wife". The first expression could be transformed into predication 'OWN' but the latter one presumably not. The classification into living and non-living objects can be used to refine the transformation to match appropriately.

3 DISCOURSE ANALYSIS WITH GRAPH TRANSFORMATIONS

In many cases it is not possible to interpret a sentence without solving references to the other sentences of the discourse. The AWARE-system makes it possible to analyze also the context of an utterance rather than only a single dependency structure. The expressions of the discourse are gathered under a single node called 'Discourse Node' (DN).

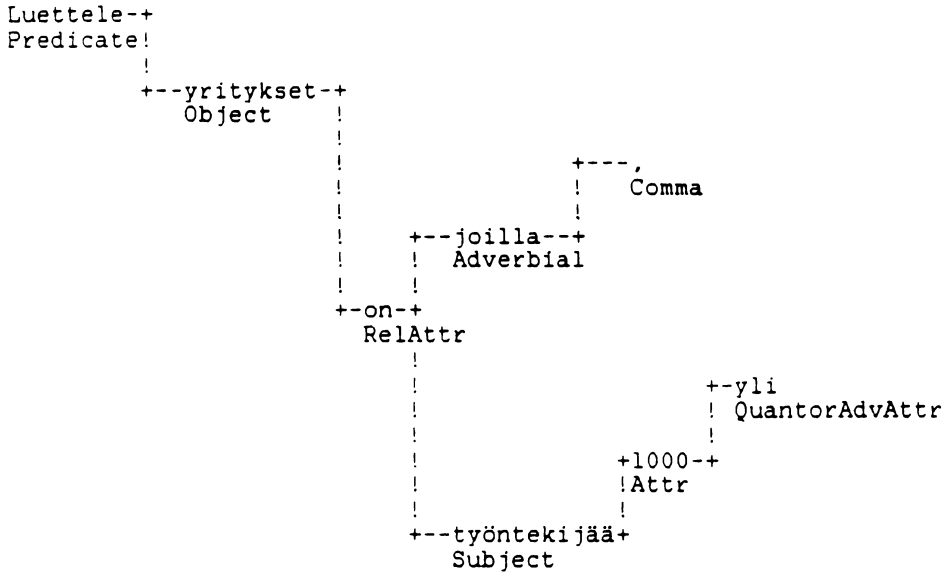


The transformations may refer to DN giving a convenient possibility to handle anaphoric and elliptic utterances.

3.1 SOLVING ANAPHORA

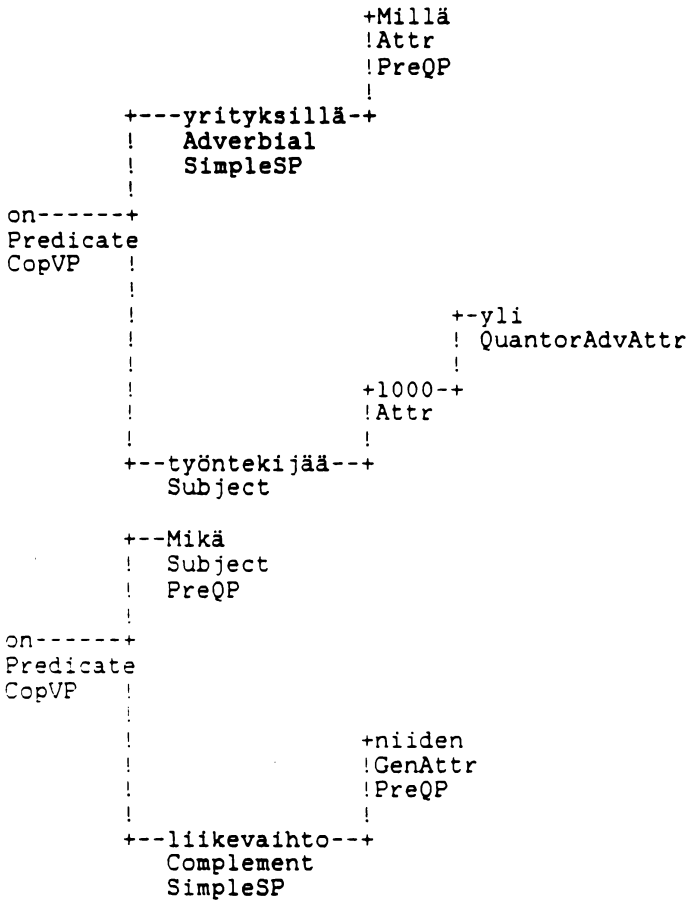
In a sentence a word may refer backwards to another word, group of words or a whole sentence replacing it. Pronouns are the most typical case. Here we give some examples of dependency structures with anaphoric reference.

(1) "Luettele yritykset, joilla on yli 1000 työntekijää!"
(Give a list of the companies that have over 1000 employees!)

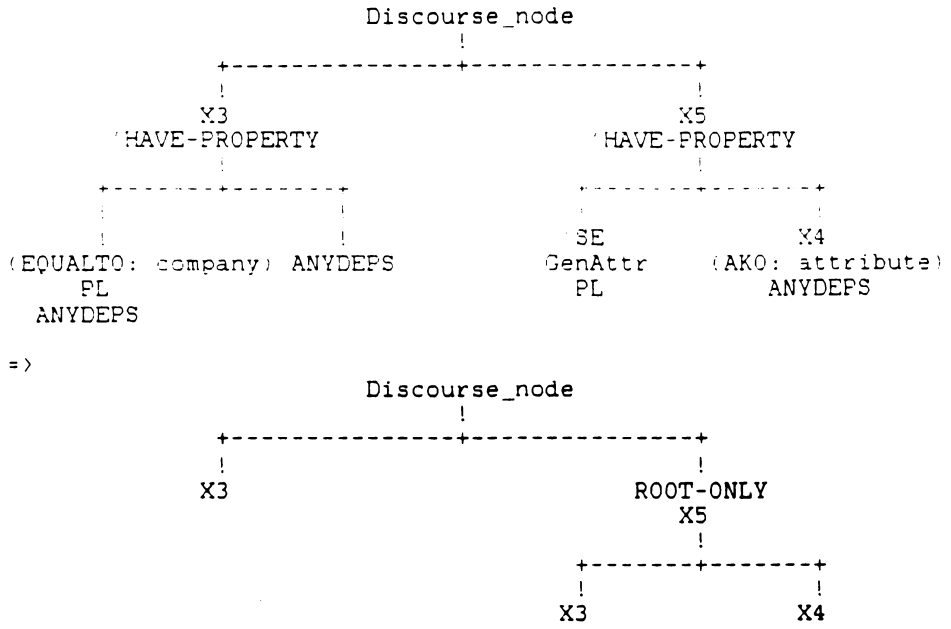


In the example a relative pronoun refers to an antecedent inside the same sentence. A pronoun may also refer to an antecedent in a previous sentence.

(2) "Millä yrityksillä on yli 1000 työntekijää?"
(Which companies have over 1000 employees)
"Mikä on niiden vuosittainen liikevaihto?"
(What is their annual turnover?)



A pair of expressions given here and especially the anaphoric reference can be analysed with the rule given below.



The node with "niiden" (plural and genetive of "se") is replaced with reference to the structure bound to variable X3.

3.2 ELLIPSIS

Often an elliptical sentence is preceeded by a complete sentence, which contains the lexical entities left out from the elliptical sentence.

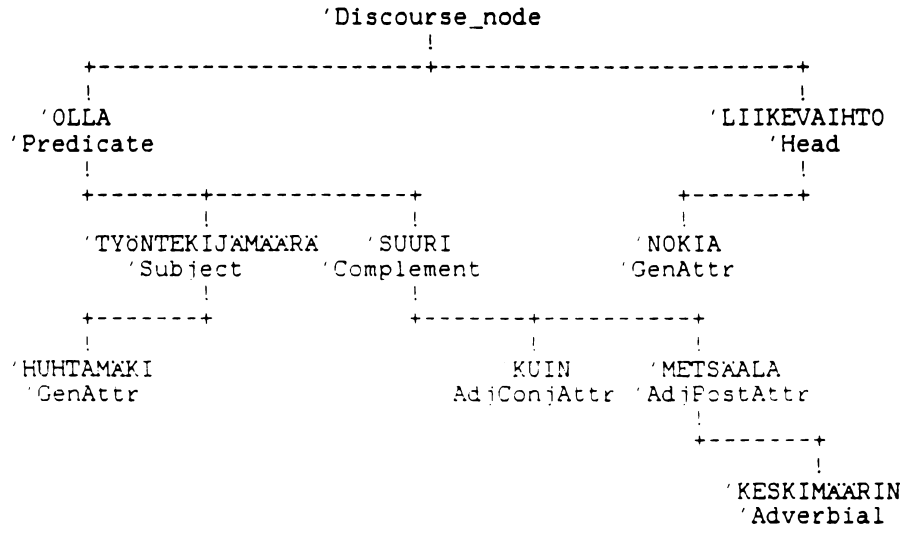
The first problem is to verify whether an expression is elliptical or not. Some heuristic rules exist but generally the decision cannot be made deterministically by analysing the sentence itself. Those heuristics might include:

- a certain expression is used together with elliptical utterance.
  ("Entäs ...","What about ...")
- the case of nominal phrase is other than nominative.
  ("Annen?", "Ann's?")
- nominal phrase is in comparative
  ("Enemmän kuin Helsingissä?","More than in Helsinki?")
- transitive verb has no object
  ("Myyty vuonna 1983?","Sold year 1983?")

The expression (1) "Turnover of Nokia?" could be understood as "Give me turnover of Nokia!". If the previous expression were (2) "Is the number of employees of Huhtamäki greater than the average in forestry" it would most propably be understood as elliptic.

Our tranformation rule example for analysis of anaphora had the discourse history as set of logico-semantic forms. AWARE-system may also be used to match parts of current expression towards the dependency structure of previous utterances.

The discourse structure of expressions (1 & 2) is shown below:



Let's consider a rule for handling an ellipsis like this. The right hand side of the rule would contain a reference from the node 'LIIKEVAIHTO' (ie. turnover) to the node 'TYÖNTEKIJÄMÄÄRÄ' (ie. number of employees) thus producing a directed acyclic graph (DAG). DAGs are usually formed when elliptical and anaphoric expressions are analyzed.

4 SEMANTIC MODELING

4.1 TRUTHCONDITIONS

The view applied in this representation is that the sense of an utterance depends on its truthconditions. This assumption is to be seen as a basis for the way how we handle the logical aspects of natural language. Though we are aware about the limitations of this approach in accordance to other

aspects of natural language.

4.2 DEPTH OF SEMANTIC ANALYSIS

The conceptual size of the domain a NLU system is developed for largely determines the semantic modeling needed. This could trivially be understood as an linear relation between the size of domain and the semantic model. Actually, if new domain areas are introduced part of the preceeding semantic modeling has to be corrected.

Let's consider again different ways of expressing 'possession'. In most of the cases 'to own' and 'to belong to' could be canonized. Compare examples below:

"This car belongs to my father"
"My father owns this car"

This general rule does not hold in all of the cases, though. Consider for example the following sentences:

"My heart belongs to my daddy"
"My daddy owns my heart"

Such examples are not just peculiarities but show the inherent character of natural language. One important consequence from this is that the methods and tools for semantic analysis should take into account these features (see e.g. Michalski, 1987) including induction and analogy. As a human being inductively infers general "rules" for her own use she also notes the exceptions for their usage.

The phenomenon known as 'the knowledge principle' in the field of artificial intelligence is analogical to the need of large amount semantic modelling for NLU systems. To get results in practical work one must have efficient tools for knowledge acquisition. The AWARE-system takes into account these needs with its graphical representation, rule generator and powerful rulebase maintenance tools. Further plans for research include development of near match analysis and use of machine learning methods.

References

Grosz,B.J.,Appelt,D.E.,Martin,P.A.,Pereira,F.C.N.:
TEAM: An Experiment in the Design of Transportable Natural-Language Interfaces. Artificial Intelligence, Volume 32, Elsevier Science Publishers, 1987, pp. 173-243.

Ishikawa,H. et al.:
A Knowledge-Based Approach to Design a Portable Natural Language Interface to Database Systems. Proceedings of the International Conference on Data Engineering, IEEE Computer Society, Los Angeles, California, 1986, pp. 134-143.

Hafner,C.D. and Godden,K.:
Portability of Syntax and Semantics in Datalog. ACM Transactions on Office Information Systems, Vol.3,No.2, pp. 141-164, 1985.

Hirst, G.:
Anaphora in Natural Language Understanding: A Survey. Springer-Verlag, Berlin 1981, 128 p.

Jäppinen, H., Honkela, T., Lehtola, A. and Valkonen, K.:
Hierarchical Multilevel Processing Model for Natural Language Database Interface. Proceedings of the 4th IEEE Conference on Artificial Intelligence Applications, San Diego, California, 1988, 6p (in print).

Lehtola, A., Jäppinen, H. and Nelimarkka, E.:
Language-based Environment for Natural Language Parsing. Proceedings of the 2nd European Conference of ACL, Geneve, 1985, pp. 98-106.

Michalski, R.S.:
How to Learn Imprecise Concepts: A Method for Employing a Two-Tiered Knowledge Representation in Learning. Proceedings of the 4th International Workshop on Machine Learning, Irvine, California, 1987, pp. 50-58.

Moore, R. C.:
Problems in Logical Form. Proceedings 19th Annual Meeting of the Association for Computational Linguistics, Stanford, The association for computational linguistics, 1981, pp.117-124.

Valkonen, K., Jäppinen, H. and Lehtola, A.:
Blackboard-based Dependency Parsing. 10th International Joint Conference on Artificial Intelligence, Milano, 1987, pp. 700-702.

NOMINALSYNTAGME-SPECIFICITET OG DISKURSUNIVERSER

Henrik Prebensen

Humanistisk edb-center
Københavns Universitet

***

1. Datalingvistik og eksperimentel lingvistik

Datamatiske systemer til forståelse af naturligt sprog vil kunne få en række praktiske applikationer, fx som grænseflader i forbindelse med databaser, ekspertsystemer, systemer til planlægning og styring af allokation af ressourcer mv.

Svagheden ved sådanne forståelsessystemer er deres relativt ringe plasticitet, dvs. evne til at klare en naturlig kommunikationssituations mange regelbrud og tvetydigheder. Interessen for dem ligger især i, at de kan spare brugerne af avancerede datasystemer for tid og besvær med at lære særlige formelle kommunikationssprog.

Opgaven er at overvinde svaghederne på en tilfredsstillende måde, så forståelsessystemer kan konkurrere effektivt med ikoniske systemer, menusystemer ol. Studiet af forståelsessystemer er derfor af klar interesse for datalingvistikkens anvendelsesmuligheder.

Dette studium har imidlertid også lingvistisk teoretisk interesse. Ved at implementere lingvistisk viden i sådanne systemer iværksætter man en form for eksperimentel lingvistik. Den kan have samme positive indvirkning på lingvistikkens udvikling, som simulation og eksperimentel kalkyle har haft og har i andre videnskaber.

Implementering af teorier, hypoteser, empirisk viden bidrager til opdagelsen af ny formalismer. Formalisering er ofte en løftestang for ny erkendelse. Implementeringen af regler mv. for konkrete sproglige fænomener giver ydermere mulighed for at evaluere gyldigheden af disse regler og rækkeviden af de begreber, de benytter. Refleksion over systemstruktur giver ligeledes mulighed for ny indsigt i sprogvidenskabelige begrebsdannelser.

Med en eksperimentel datamatisk implementering af et grammatisk fænomen igangsættes en dialektisk proces mellem afprøvning af hypoteser, afklaring af begreber, udvidelse af problemstillinger og sammenkobling af teorier og vidensområder, en proces, der kan være af stor gavn for forståelsen af sprog og tænkning i almindelighed. Med datamaten som eksperimentelt værktøj står lingvistikken i en ændret forskningssituation.

2. Blockhead eksperimentet

Dette foredrag bygger på et arbejde med et eksperimentelt lingvistisk forståelsessystem skrevet i PC-PROLOG (alias TURBO-PROLOG). Eksperimenterne vedrører betydning og forståelse af nominalsyntagmer, der refererer til genstande i en ydre virkelighed.

Det eksperimentelle system er en simpel klodsverden, hvori en robot, *Blockhead*, ved hjælp af en krog kan flytte rundt på en lille samling klodser på et bord og besvare spørgsmål om klodsverdenens tilstand. Tilstanden vises i et vindue på en dataskærm. Ordrer og spørgsmål til *Blockhead* skrives i et felt under vinduet, hvor også *Blockheads* svar fremkommer. Se i øvrigt skærmbillederne i appendix.

Der gælder de samme vilkår for *Blockhead* som for alle eksperimenter: begrænsningens kunst er vigtig. Eksperimentet uvedkommende omstændigheder må skrælles bort eller reduceres til det minimale. Det væsentlige er, overskueligt og uden "støj" fra parasitfænomener, at kunne manipulere de forhold, eksperimentet er sat op for at studere: i *Blockhead* nominalsyntagmers referenceforhold.

Blockheads verden omfatter derfor kun 8 genstande. Genstandene har kun 3-4 egenskaber. *Blockheads* sprog er et begrænset engelsk, selv om antallet af sætninger og perioder, robotten kan forstå, er (rekursivt) uendeligt.

Ordforrådet omfatter mindre end 40 indholdsord (verber, nominer, adjektiver) og under 70 "grammatiske" ord (pronominer, præpositioner, adverbier, konjunktioner).

Morfologisk analyse er reduceret mest muligt. Fx kan kun præsens indikativ og imperativ, samt enkelte sammensatte former af verber, fx passiv eller *extended form (-ing)* bruges. Nominalsyntagmer optræder kun i singularis, osv.

Det egentlige hovedformål med programmet er at studere **anafori.** De fænomener, der gives en uddybet behandling, er derfor sådanne som spiller en rolle herfor, nemlig nominernes omfangsbetydning eller ekstension, deres bestemthed, deres bestemmelser (relativsætninger, præpositionsforbindelse, mv.), deres optræden i komplekse perioder med sideordnede og betingede sætninger, i spørgsmål og ordrer.

Det er i et sådant arbejde fristende at gøre programmet mere "intelligent", og dermed mere overbevisende og publikumsvenligt ("sexy"). Komplikationer er imidlertid kun berettigede, hvis de er teoretisk interessante. Hensigten med *Blockhead* er ikke at imponere med hensyn til, hvad en datamat kan fås til at gøre, men at øge lingvistens forståelse af sprogets mekanismer, afklare sprogvidenskabelige begreber, gennemprøve formelle værktøjer.

Blockhead-eksperimentet indgår i et projekt: *Anaforisk resolution,* støttet af *Statens humanistiske forskningsråd* under FTU-bevillingen. Projektet er treårigt (1986-88) og foregår ved *Humanistisk edb-center, Københavns Universitet.*

3. Problemet om specifik reference

Det problem jeg her ønsker at diskutere i relation til *Blockhead*-eksperimentet er problemet om nominalsyntagmers specifikke reference: specificitetsproblemet. Specifikke nominalsyntagmer refererer til identificerbare størrelser (entiteter) i et univers; det gør non-specifikke nominalsyntagmer ikke:

(1) a. Marie vil giftes med Ola
(1) b. Marie vil giftes med en nordmand

I (1)a. har *Ola* specifik betydning. Der findes et individ med dette navn, som et andet individ benævnt *Marie* vil ægte. (1)b. har derimod to betydninger afhængigt af *nordmands* specificitet: der kan som i (1)a. være tale om et specifikt individ, som *Marie* vil ægte. Men der kan også være tale om en type. *Nordmand* beskriver da en egenskab ved en ægtemand som *Marie* kunne ønske sig. Der refereres ikke til et specifikt individ, som findes her og nu.

Ved siden af specifik, non-specifik, anvendes andre (næsten) synomyme betegnelser: *transparent/opaque, de re/de dictu*. Fra Frege kendes også betegnelserne *Bedeutung* og *Sinn*, fra logikken *ekstensional* og *intensional* betydning.

Betydningen af et specifikt nominalsyntagme opfattes i moderne semantik som dets *ekstension* eller begrebsomfang, dvs. den mængde af referenter, syntagmet kan udsiges sandt om i et bestemt, givet univers: den aktuelle verden, tekstens virkelighed, diskursuniverset eller hvad man vil kalde det.

Et non-specifikt nominalsyntagmes betydning ses derimod som *intension* eller begrebsindhold; det kan opfattes som refererende til en proces eller metode eller et sæt betingelser, der kan bruges til at bestemme ekstensionen i et hvilket som helst univers. Dvs. betydningen er et sæt af virtuelle ekstensioner: I (1)b. med non-specifik fortolkning skal prædikatet *nordmand* være sandt om det individ, der i en af de fremtidige (mulige) verdener repræsenterer *Maries* ægtemand.

For at undgå at operere med verdener som stedet for nominalsyntagmers ekstension, hvilket kan give mængdeteoretiske problemer, kan man operere med begrænsede "verdenstilstande" eller mulige situationer. Således kan forskellen mellem (2)a. og b.

(2) a. Præsidenten i Santa Marihuana bliver myrdet i dag
b. Præsidenten i Santa Marihuana bliver myrdet lidt for ofte

beskrives ved at sige, at i (2)a. er ekstensionen dagens præsident i *Santa Marihuana*, mens der i (2)b. itereres over et sæt af tilstande, sådan at *præsident i Santa Marihuana* kan udsiges med sandhed om den relevante referent i hver af tilstandene.

Det specifikke nominalsyntagme har sin ekstension i en her- og nu-tilstand defineret ved et bestemt sæt tids- og stedskoordinater. Det non-specifikke har ikke ekstension i en sådan veldefineret tilstand. Derfor forekommer non-specifikke nominalsyntagmer ved modalverber og andre modale udtryk, ved nægtelse, ved iterative udtryk (herunder visse kvantorer), attitudeudtryk, imperativer og spørgsmål. Fælles for disse er, at de peger bort fra den **aktuelle** situation, mod en eller flere **virtuelle** situationer, der er mulige

(modale udtryk), forestillede (imperativer, attitudeudtryk), ukendte (spørgsmål), eller ikke-eksisterende (negation). Skal der ske en anaforisk genoptagelse af non-specifikke nominalsyntagmer i en diskurs, skal der tales i en virtuel modus. Derfor kan anaforer virke baglæns disambiguerende ved tvetydig specificitet:

(3) a. Marie vil giftes med en nordmand. Han hedder Ola.
b. Marie vil giftes med en nordmand. Han skal være stor, lys og blåøjet.
c. Marie vil giftes med en nordmand. Han skal være stor, lys og blåøjet. Han må gerne hedde Ola. ?* Han står derovre.

I (3)a. disambiguerer *han* syntagmet *en nordmand:* ekstensionen må være i den aktuelle verden, fordi fortsættelsen ikke er markeret som virtuel. I b. er fortsættelsen modalt markeret, så ekstensionen henlægges til en fremtidig tilstand. I c. sker der et brud. *Han* optræder først som i b., derefter som i a. Hvis ekstensionen både skal være i den aktuelle og en virtuel tilstand, må diskursen fortolkes spidsfindigt, fx sådan at den talende kender *Maries* fremtid, men at *Marie* ikke selv kender den.

Hvorledes kan et datamatisk sprogforståelsessystem simulere en sådan forståelse af nominalsyntagmer, dvs. hvordan kan den formaliseres i en teori for fænomener som specificitet?

4. Definition af forståelsessystemer

Et semantiske forståelsessystem, som det i Blockhead anvendte, kan formelt defineres som en modelteoretisk struktur, S:

(4) $S = <L, M, F, N, T>$

hvor **L** et formelt sprog (det semantiske repræsentationsprog),
M er en mængde af entiteter eller genstande (modellen),
F en afbildning, $F: L \rightarrow M \cup \{0,1\}$, (interpretationsfunktionen),
N en mængde af sætninger og perioder i et naturligt sprog,
T en automat, der beregner en afbildning, $T: N \rightarrow L$.

L, det semantiske repræsentationssprog (den semantiske repræsentation), er defineret over et vokabular, **V**, der består af symboler for *relationer* (prædikater), *funktorer* og *termer.*

Hver *relation* og hver *funktor* har en "aritet", der angiver antallet af dens argumenter.

En *term* er et enkelt symbol eller en streng af symboler. En *term* kan være en *konstant*, en *variabel* eller *et komplekst udtryk indledt af en funktor.*

En *velformet formel* (syntaktisk korrekt udtryk) i **L** er en liste, der som hoved har et relationssymbol og som hale en liste af termer, i antal svarende til relationens aritet. Et sådant udtryk kaldes en *proposition.*

En *proposition* er *grundet,* hvis den kun indeholder konstanter, *ugrundet,* hvis den indeholder blot 1 variabel eller funktor.

M, modellen, er det semantiske repræsentationssprogs domæne, dvs. den mængde af genstande, hvori relationerne og termerne i **L** har ekstension.

Man kan i et datamatisk system forestille sig **M** som en *database*, hvor entiteterne er poster eller stamkort, der hver bærer et index eller et navn som identifikator. De egenskaber ved og de relationer mellem entiteterne, der er relevante i en given applikation, repræsenteres da af statiske databaseprædikater, og en given tilstand af modellen repræsenteres af en mængde af databaseklausuler. En forandring, hvorved modellen dynamisk går fra en tilstand til en anden, manifesteres ved, at en given databaseklausul slettes (fx med *retract*), og en anden indsættes (med *assert*). Forandringer sker i overenstemmelse med regler, der indeholder betingelser for sletning og indsættelse.

F, interpretationsfunktionen eller fortolkeren, er en kompleks afbildning, der har udtryk i **L** som definitionsmængde og ekstensioner i **M** eller {**1,0**},(dvs. {**sand,falsk**}) som billedmængde.

F består af en tilskrivningsfunktion, der tager en term i **L** som argument og har en entitet i **M** som værdi og af en evalueringsfunktion, der tager en grundet proposition fra **L** som argument og undersøger, om den er konsistent med modellens tilstand.

Evalueringen afhænger af propositionens type. Hvis propositionen er en kommando, undersøges det om der findes et par af udsagn, der beskriver modellens tilstand før og efter forandringen, og som respekterer en regel for forandring. Hvis propositionen er et udsagn, undersøges dets sandhedsværdi i forhold til modellen.

En proposition, der er udførbar eller sand, kaldes *modelkonsistent*. **F** evaluerer altså *modelkonsistensen* for propositioner i **L**.

F er implementeret som en tilbagesporende proces, der først instantierer variabler og andre ugrundede argumenter for at frembringe en grundet proposition, hvis modelkonsistens så evalueres. Hvis en instantiering viser sig *modelinkonsistent*, sker der tilbagesporing. Således afprøves alle alternative instantieringsmuligheder, inden modelinkonsistens accepteres.

N er en (i princippet uendelig) mængde af sætninger i et "naturligt" sprog, fx *Blockhead-engelsk*.

T er en transducer, dvs. en automat der "oversætter" sætninger i **N** til de semantiske repræsentationer i **L**.

T omfatter som minimum et leksikon, et sæt syntaktiske dekompositionsregler og et sæt semantiske kompositionsregler. **T** foretager en niveaudelt syntagmatisk analyse af en inputsætning til ord- eller morfem-niveau. **T** finder betydningsrepræsentationerne (i **L**) af ord/morfemer i leksikon. **T** danner syntagmernes betydningsrepræsentationer (i **L**) ud fra konstituenternes betydninger og den syntaksregel, der konstituerer hvert syntagme (cf. Freges kompositionsprincip).

I hver regel i **T** foregår der altså på en gang en syntaktisk analyse af et syntagme og en semantisk syntese af syntagmets betydning på basis af de udanalyserede konstituenters betydninger.

Hvis systemet implementeres i PROLOG, kan **T** benyttes i begge retninger, dvs. **T** kan også tage en streng i **L** som input og syntetisere den streng i **N**, der er det natursproglige svar på et spørgsmål.

Den minimale transducer er imidlertid utilstrækkelig til en rimelig simulering af forståelsen af naturligt sprog. Fx kan betydningen af anaforiske udtryk, såsom pronominer, ikke findes ved hjælp af et leksikon. **T** må derfor udvides med en anaforisk proces, der tillader at gemme og hente sådanne betydningsrepræsentationer, som er afhængige af konteksten.

Endvidere er det hensigtsmæssigt at lade **T** benytte **F** til at teste værdier, der tilskrives ugrundede termer *on the fly*, dvs. mens et syntagme analyseres, men inden hele sætningen er analyseret, for at **T** på denne måde altid hurtigst muligt kan give en *grundet* proposition som output.

En fordel ved denne strategi er, at man ved straks at lede efter en konstant som repræsentation for et nominalsyntagme og teste den for konsistens med modellens tilstand undgår *kombinatorisk eksplosion* beroende på syntaktisk flertydighed af nominalsyntagmer. **T** vil altid søge tidligst muligt at finde én og kun én grundet, modelkonsistent repræsentation af input.

T og **F** kan dele informationer om analysen og den semantiske interpretation ved at skrive eller læse på den samme tavle. Tavlen ændrer ikke systemet formelt. Det ville være muligt at undvære den og i stedet overføre alle de relevante informationer som parametre. Ulempen herved er rent teknisk: lange parameterlister og mange tilfælde, hvor parametrene er tomme.

Den her skitserede implementering er teoretisk tilfredsstillende, fordi den giver et formelt veldefineret indhold til mange semantisk-pragmatiske begreber, bl. a. *anafori* og *specificitet*.

På basis af formalismen kan vi nu definere begrebet *forståelse* i forhold til et system ved at sige, at en sætning *P* i **N** forstås af et forståelsessystem, **S**, hvis og kun hvis **T** kan generere en repræsentation af *P* i **L**.

Hvis **S** ikke forstår *P*, kan det skyldes, at *P* er ikke et tilladt input for **T**, fordi *P* er en ukorrekt sætning, eller fordi *P* ikke vedrører det univers, **S** er konstrueret til, fx *Blockheads* klodsverden. Det kan også være at *P* krænker en præsupposition, fx unicitetspræsuppositionen, der omtales nedenfor. Et system som det her definerede vil være i stand til at informere brugeren om grunden(e) til sådanne forståelsesvanskeligheder.

I det følgende beskrives dele af en konkret udformning af et sådant system med *Blockhead* som eksempel.

5. Den semantiske repræsentation, L

L skal kunne repræsentere de tre fundamentale sproghandlingstyper: ordre, spørgsmål og beskrivelse. I gængs logisk semantik har den deklarative sproghandling altid været anset som den fundamentale. Spørgsmål og ordrer behandles ofte slet ikke. Der er imidlertid mange fordele ved at tage de imperative og interrogative typer som primære, dvs. lade forståelsen af dem være forudsætning for behandlingen af den deklarative.

En velformet formel i **L** er en liste med et prædikat som hoved og en række argumenter som hale. Den repræsenterer en sætning i **L**. Hvilken funktion (imperativ, interrogativ, deklarativ) sætningen har, repræsenteres som dens *modalitet.*

(5)	a.	N: put the white block into the box!	
		L: [move, whiteblock, box]	modalitet(!)
	b.	N: is the white block on the table?	
		L: [stat, whiteblock, table]	modalitet(?)
	c.	N: where is the white block?	
		L: [stat, whiteblock, x]	modalitet(?)
	d.	N: which block is situated in the box?	
		L: [stat, xblock, box]	modalitet(?)
	e.	N: there is a block in the box.	
		L: [stat, ablock, box]	modalitet(.)

move og *stat* er relationer, der beskriver henholdsvis forandring og tilstand.

Argumenterne er termer: *whiteblock, box, table* er konstanter; *x, xblock, ablock* er variabler. a.-b. er derfor grundede, c.-e. ugrundede.

Der er 3 slags variabler: den generelle variabel, *x,* der repræsenterer de "totale" spørgeord *what, which, where;* den typologiserede spørgende variabel, *xblock,* der repræsenterer nominalsyntagme med spørgende determinativ, *which block, what block*; den typologiserede indefinite variabel, *ablock*, der repræsenterer nominalsyntagme med ubestemt determinativ, som *a block, some block, any block.*

Der opereres med flere slags variabler, - modsat hvad der tilfældet i semantikker baseret på deklarativ sprogbrug - af hensyn til den korrekte behandling af spørgsmål. Disse deles som bekendt normalt i *helspørgsmål* og *delspørgsmål.* Helspørgsmål indeholder ikke spørgeord og kan besvares med *ja/nej.* Delspørgsmål indeholder spørgeord og kan ikke besvares med *ja/nej,* men med et syntagme, der indsat på spørgeordets sted verificerer udsagnet: *Hvilken klods står i kassen?* - **Den gule**.

Spørgsmål med ubestemt nominalsyntagme er helspørgsmål, men de har en vis lighed med delspørgsmål. De kan besvares med *ja/nej,* men i tilfældet *ja* oftest suppleret med et svarsyntagme, der erstatter det ubestemte nominalsyntagme, ligesom svarsyntagmet erstatter det spørgende syntagme ved delspørgsmål. Et rent *ja*-svar vil i modsat fald ofte afføde et delspørgsmål for at få den supplerende oplysning: *Står der en klods på bordet? - Ja.* - **Hvilken klods? - Den gule**. Svaret kunne derfor lige så godt lyde: **Ja, den gule.** Disse spørgsmål kan derfor kaldes *partielle helspørgsmål.*

For at der kan generes korrekte svar på alle tre slags spørgsmål, må den semantiske repræsentation kode information om spørgsmålets type. Ved det rene helspørgsmål skal sandhedsværdien af et grundet udsagn bestemmes. Ved de andre spørgsmålstyper, hvor der instantieres en variabel, skal der også gives information tilbage om, hvilken instantiering, der har været brugt til at give værdien *sand.*

At de typologiserede variabler noteres som strenge, er en ren notationskonvention. De kunne være noteret "polsk", med funktorer, fx *block(x).*

(5)a.-b. er grundede. *Blockhead* kontrollerer, om konstanterne er navne på entiteter i **M**, og om relationen er modelkonsistent for dem.

(5)c.-e. er ikke-grundede. Her skal fortolkeren **F** prøve at instantiere de variabler med konstanter, der er navne på entiteter, som gør prædikatet modelkonsistent.

Ved nominalsyntagmer, der er definite eller indefinite beskrivelser *(the block on the hook, a block on the table)*, genererer **T** en term med en funktor, *IDENTIFY*, som hoved og en proposition i **L** som hale.

(6) a. N: pick up the block in the box!
L: [move, IDENTIFY stat xblock box, hook] modalitet(!)
b. N: place the box on a block on the table!
L: [move, box, IDENTIFY stat ablock table] modalitet(!)
c. N: find a block which is situated on a block on the table!
L: [regard, IDENTIFY stat ablock IDENTIFY stat ablock table]
modalitet(!)

Propositionen efter funktoren *IDENTIFY* er af praktiske grunde noteret som en streng. Den konverteres af fortolkeren **F** til et spørgsmål, og svaret på spørgsmålet indsættes som konstant i den overordnede proposition.

Altså i (6)a. evaluerer **F** *[stat, xblock, box]* som et spørgsmål. Hvis fx *yellowblock* verificerer propositionen, erstattes termen *IDENTIFY stat xblock box* med *yellowblock*. Tilsidst evalueres sandhedsværdien af den derved fremkomne grundede proposition i modellen. Tilsvarende med (6)b.

I (6)c. er der en rekursiv indlejring af termer med funktoren *IDENTIFY*. **F** vil først evaluere den inderste proposition, *[stat, ablock, table]*. Lad os sige, at *whiteblock* verificerer den. Nu evalueres den ydre proposition med *whiteblock* som andet argument: *[stat, ablock, whiteblock]*. Lad *yellowblock* verificere den. Sidst evalueres *[regard, yellowblock]*.

I **L** er termerne altså konstanter, der er navne på entiteter i **M**, fx *whiteblock, box*, variabler af tre slags: *x, xblock, ablock*, der instantieres af **F**, eller funktorudtryk med *IDENTIFY* som hoved, hvor halen evalueres som spørgsmål.

6. Behandlingen af unikke nominalsyntagmer i T

Transduceren **T**s opgave er at generere de korrekte repræsentationer i **L** for nominalsyntagmerne i **N**. Herunder bruger **T** den strategi, at variabler i propositionelle udtryk tidligst muligt skal erstattes med konstanter. **T** søger altså altid at generere en tilfredsstillende *grundet* repræsentation i **L** af inputsætningerne i **N**.

Desuden arbejder **T** ud fra den strategi, at afsenderen altid har gjort sit bedste for at sige noget meningsfuldt, dvs. at **T** på enhver måde skal prøve at forstå (finde en repræsentation af) det sagte.

Transduceren **T** behandler nominalsyntagmer efter to hovedregler: reglen for *unica* og reglen for *non-unica*.

Begrebet *unicitet* betegner en præsupposition vedrørende et nominalsyntagmes ekstension. Denne præsupposition markerer afsenderen med determinativet. Det *bestemte* determinativ signalerer, at referenten er et *unicum*, enestående i modellen, og det forventes, at modtageren kan bruge denne information til at identificere referenten.

Denne type af *unicitet* kaldes *referentiel unicitet.* Hvis der fx i klodsverdenen er én og kun én kasse, omtales den som *the box.* Hvis der er flere, kan ingen omtales som *the box.* Hvis der er én og kun én klods, der er hvid, kan den omtales som *the white block.* Tilsvarende for *the block on the hook, the block on the table which supports the box,* osv.

Referentiel unicitet har været en del diskuteret i logik i forbindelse med Bertrand Russells *theory of definite descriptions.* Problemet er, om brud på uniciteteten giver meningsløse eller falske udsagn: *the present king of France is bald* versus *the present king of France is not bald.* Russell hævder, på grundlag af sin teoris definition af *bestemthed,* at begge udsagn er falske. I *Blockhead* er de meningsløse.

En fejlagtig præsupposition vil i *Blockhead* blive opdaget af **T.** Hvis afsenderen anvender udtryk som *the block which is situated on the table* (eller *the block on the table*), oversættes det, som vi har set, med *IDENTIFY stat xblock table.* Når spørgsmålet *[stat, xblock, table] (which block is situated on the table?)* evalueres, og der står flere klodser på bordet, kan **T** ikke give et svar. Spørgsmålet præsupponerer nemlig unicitet. Fortolkeren, **F**, konstaterer, at der er mere end én klods, der verificerer udsagnet og kan derfor gøre opmærksom på, at spørgsmålet er stillet med forkerte forudsætninger.

F er altså i stand til at undersøge, om en referentiel unicitet er forudsat ved brug af et delspørgsmål. Denne egenskab bruger **T** på adnominale syntagmer som fx relativsætninger eller præpositionssyntagmer med propositionel værdi.

Foruden referentiel unicitet findes *anaforisk unicitet. The block* kan som ekstension godt have en bestemt klods, selv om der ikke foreligger en unik klods i modellen, nemlig hvis den pågældende klods har været omtalt og genoptages anaforisk: *if there is a block in the box, then pick up* **the block!** På samme måde har bestemte pronominer som *it, this* og *that* anaforiske unica som ekstension.

Reglen for unica aktiverer i *Blockhead* en proces, så snart **T** møder et bestemt determinativ. Processen forsøger at læse så meget af den efterfølgende streng som nødvendigt for at identificere det korteste nominalsyntagme, der har et unicum som ekstension. **T** tester herunder hele tiden for såvel referentiel som anaforisk unicitet.

Denne proces er særdeles betydningsfuld ved afgørelse af syntaktisk flertydighed. I sætningen

(7) Put the block in the box on the block on table!

er der syntaktisk set mulighed for flere afgrænsninger af syntagmerne, fx

(7) a. the block // in the box on the block on the table
(7) b. the block in the box // on the block on the table
(7) c. the block in the box on the block // on the table

Præpositionerne *in* og *on* kan nemlig begge være både verbalafhængige (afhænge af verbalet *put*) og relationer i et udsagn *([stat, xblock, box]).* Flertydigheden undgås, hvis der bruges en entydigt verbalafhængig præposition, som *onto*

(7) d. Put the block in the box onto the block on the table!

Når flertydigheden i de fleste tilfælde alligevel ikke erkendes, kan det skyldes, at referenterne er unikke i situationen, og denne (med bestemt artikel signalerede) unicitet, redder entydigheden. Modtageren kan nemlig benytte den til at løse flertydigheden med

det samme, dvs. når nominalsyntagmernes afgrænsninger undersøges. Forståelsen af et nominalsyntagme er altså delvis "lokal".

I et tilfælde som dette, vil **T** først prøve om *the block* alene opfylder unicitetsbetingelsen, fx er anaforisk unik.

Hvis det ikke er tilfældet, vil **T** forsøge om *the block in the box* har en unik referent.

Hvis det heller ikke er tilfældet, forsøges med syntagmet *the block in the box on the table*.

På denne måde undgår systemet den kombinatoriske eksplosion, som en rent syntaktisk analyse ville give.

Uniciteten gør det også muligt at løse tvetydigheder mellem restriktive og parentetiske relativsætninger:

(8) N: Pick up the pyramid which is on the table!
L1: [move, IDENTIFY stat xpyramid table, hook] (restriktiv)
L2: [move, pyramid, /[stat, pyramid, table]?/, hook] (parentetisk)

Ts unicitetsprocedure undersøger først om *the pyramid* er referentiel eller anaforisk unik.

Hvis ingen af delene er tilfældet (L1), samtidig med at unicitetspræsuppositionen siger, at der skal være en unik referent, må informationen i den efterfølgende del af strengen være "nødvendig" for at identificere unicum. Derfor genereres en *IDENTIFY* struktur, som **F** senere behandler som spørgsmålet: *which pyramid is placed on the table?*. Svaret udgør den eftersøgte unikke referent.

Hvis *the pyramid* er referentiel eller anaforisk unik (L2), kan der ikke i reststrengen være indeholdt information, der er nødvendig for identifikationen af en unik referent. Hvis der derfor optræder en relativsætning, kan den kun være parentetisk. Den behandles derfor som et helspørgsmål: *is the pyramid placed on the table?,* der forventeligt skal besvares med *yes,* hvis ikke relativsætningen skal være nonsens.

Sammenfattende kan det siges, at unicitetsanalysen hviler på muligheden af at bruge såvel morfologisk som syntaktisk, semantisk og pragmatisk information til at afgrænse det bestemte nominalsyntagme lokalt, dvs. uden hensyntagen til det overordnede strukturniveau.

Strategien bygger på, at substantivagmet altid begynder med et bestemt determinativ til venstre. Derefter følger en substantivisk kerne, der evt. kan foregås af et adjektivsyntagme:

(9) $[_{np}$ the ($[_{ap}$ small black]) $[_{n}$ block] ...

Kernesubstantivet kan syntaktisk set være højreafslutningen på hele syntagmet. Det, der testes ved unicitetsproceduren er, om denne afgrænsning giver mening i konteksten, altså om der findes en unik referent eller anafor til *the (small black) block.* Den semantiske evaluering sker på stedet ved hjælp af fortolkerens spørgemekanisme.

Hvis svaret på fortolkerens spørgsmål er *ja*, må resten være et syntagme med funktion på det højere niveau (verbalafhængigt fx) eller en parentetisk udvidelse til *the (small black) block*, svarende til et indskudt udsagn, dvs. noget som kan evalueres som et helspørgsmål.

Hvis svaret er *nej*, må en del af den efterfølgende streng skulle medinddrages i syntagmet før højregrænsen kan sættes. T gnaver sig derfor frem til næste potentielle syntagmegrænse og evaluerer den derved fremkomne syntese i forhold til modellen for at se, om grænsen ligger der:

(10) $[_{np}$ the $([_{ap}$ small black$])$ $[_{n}$ block$]$ $[_{restr}$ in the box$]$ ...

Således fortsættes, så længe der ikke er fundet en meningsfuld substantivisk helhed og der stadig til højre er en streng, der syntaktisk kan være en del af et substantivsyntagme.

7. Behandlingen af non-unikke nominalsyntagmer i T

Hvis et substantivsyntagme er indledt af ubestemt determinativ, er der ikke gjort nogen antagelse om unicitet. Dvs. modtageren er frit stillet i sine fortolkningsmuligheder. Der vil normalt være tale om mange mulige referenter, selv om der godt kan være tale om kun én. Det afgørende er, at afsenderen ikke giver nogen information om forudsætningerne, og at modtageren derfor frit må vælge en referent inden for de givne muligheder.

Når T læser et ubestemt nominalsyntagme

(11) $[_{np}$ a $([_{ap}$ black$])$ $[_{n}$ block$]$ ...,

kan T imidlertid ikke på basis af en lokal semantisk syntese foretage et valg. Dels kan der i det efterfølgende være udvidelser til syntagmet med informationer, der begrænser valgmulighederne. Dels kan valget være begrænset af kravet om, at hele sætningen eller hele den periode, sætningen indgår i, skal være meningsfuld, dvs. modelkonsistent. T genererer derfor en midlertidig repræsentation af syntagmet (en variabel), der tillader at udsætte valget.

Hvis der fx er tale om sætningen

(12) move a block into the box!

nytter det ikke at instantiere med navnet på en klods, der senere viser sig ikke at kunne være i æsken. Derfor genereres variabelen, *ablock*, der senere instantieres af fortolkeren *med tilbagesporing*, indtil der opnås en grundet, modelkonsistent ordre.

Hvis der fx er tale om perioden

(13) pick up a block and put it into the box and place the pyramid on it!

kan det ikke nytte at *a block* instantieres med navnet på en klods, der er for stor til at være i æsken og for lille til, at pyramiden kan stå på den. Valget er begrænset til den delmængde af de disponible klodser, der opfylder de rette krav. Dette problem løses i *Blockhead* med et "fantasi-modul", der er en del af fortolkeren.

Fantasi-modulet opretter en stak af virtuelle tilstande med de forandringer, der svarer til en række kommandoer. På den måde sker der forward-chaining frem mod målet: en instantiering, der gør alle sætningerne i en periode modelkonsistente.

I (13) vil fantasimodulet først gemme den aktuelle tilstand. Nu vil **T** søge en repræsentation til den efterfølgende sætning. Fantasimodulet vil oprette en ny tilstand, der svarer til den beordrede forandring, og stakke den ovenpå den første, og derefter fortsætte således til sidste sætning. Hvis det undervejs viser sig, at en proposition ikke er modelkonsistent med den sidst stakkede tilstand, spores der tilbage gennem stakken til sidste tilstand, i hvilken der fandtes en alternativ instantiering. På denne måde foregår der en søgning igennem et træ af instantieringer, indtil der er fundet en modelkonsistent serie af repræsentationer af kommandoerne, eller indtil alle muligheder er udtømt.

Hvis der er en udvidelse, fx relativsætning, til et ubestemt nominalsyntagme:

(14) a. N: a block which supports the white one

genereres en *IDENTIFY* term svarende til et partielt helspørgsmål:

(14) b. L: IDENTIFY stat-invert ablock whiteblock

F evaluerer (14)b. som spørgsmålet *does any block support the white one?* og indsætter den konstant, der indgår som svar, fx *yes - the big black block* - på *IDENTIFY*-termens plads.

Det afgørende ved den non-unikke analyse af ubestemte nominalsyntagmer er altså, at modtageren kan vælge mellem flere instantieringer af syntagmet med konstanter, men at valget er bundet af betingelser, som sikrer, at den resulterende proposition er modelkonsistent.

Strategien for **T** er at repræsentere non-unikke nominalsyntagmer med variabler, der instantieres tidligst muligt ved kald af **F**, evt. under anvendelse af fantasi-modulet, der gennemprøver alle alternativer.

8. Anaforer

Der opereres med to slags anaforer, *N_anaforer* og *NP_anaforer.*

N_anaforen er det sidst mødte nomen, som transduceren **T** gemmer på systemets tavle. Det benyttes løbende som opslagsord for pronominer som *one* i udtryk som *the white one.*

NP_anaforerne er en liste af de konstanter, der har optrådt som argument-hale i den sidst forudgående proposition. Det er **F**, der sørger for, at de gemmes. De benyttes af **T** til at instantiere pronominer i den næstfølgende sætning.

I det første tilfælde er det en leksikalsk information, der gemmes, i det andet en semantisk. NP_anaforen er ikke (undtagen for så vidt angår genus) afhængig af formen på antecedenten, men af betydningen, af antecedentens extension.

NP_anaforerne er altså extensioner i modellen, som etableres løbende på basis af det sagte. Gives der en ordre

(15) Put the yellow block onto the block in the box on the table!

er anaformængden efter ordrens evaluering entiteterne *yellowblock* og *whiteblock*, dersom *the block in the box on the table* kan instantieres med denne.

9. Diskursunivers og specificitet

Vi kan nu definere begreberne *diskursunivers* og *specificitet* med reference til et modelteoretisk forståelsessystem som *Blockhead.*

> Lad der være givet et modelteoretisk system S.
>
> En sekvens af perioder, $(P_1,...,P_n)$, hvor hver periode består af 1 til m sætninger i N, kaldes *en diskurs.*
>
> Ved en *NP_anafor* i en diskurs vil vi forstå en konstant, der benævner en entitet i modellen, M, og som er forekommet som argument i en proposition, der repræsenterer den sidst forståede sætning i diskursen.
>
> Ved *diskursuniverset* vil vi forstå unionen af modellen og den til enhver tid givne mængde af NP_anaforer.

Herefter kan vi definere begreberne *specifik* og *non-specifik* således:

> Et nominalsyntagme er *specifikt*, hvis det af transduceren, T, repræsenteres med en konstant.
>
> Et nominalsyntagme er endvidere *specifikt*, hvis det af T repræsenteres af en variabel, og af fortolkeren, F, instantieres med en konstant i en modelkonsistent proposition.
>
> Et nominalsyntagme er *non-specifikt*, hvis F ikke kan erstatte det med en konstant.

Disse definitioner bygger på processer, som udføres af et modelteoretisk system, *in casu* systemet *Blockhead.* De kan imidlertid benyttes som eksperimentalteoretiske forklaringer på nogle af egenskaberne ved de tilsvarende fænomener i naturlige sprog. Hermed menes, at disse egenskabers eksistens nu kan udledes som konsekvenser af egenskaber ved systemet, og deres fordeling forudsiges på grundlag af fordelingen i systemet.

For det *første* kan vi nu ud fra systemet forklare, hvorfor bestemte nominalsyntagmer er *specifikke.* Bestemthed er i *Blockhead* implementeret som en proces, der forudsætter at afsenderen ved, hvilken entitet han refererer til. Da han med determinativet markerer en præsupposition om, at referenten er unik i diskursuniverset, følger at han faktisk ved, hvilken entitet han taler om. Den pågældende entitet er derfor specifik *per se.*

For det *andet* kan vi nu ud fra systemet give mening til begerebet *non-specificitet* og forklare den tvetydighed, der er knyttet dertil (se § 3). Et ubestemt nominalsyntagme er *ikke* markeret for *unicitet* og derfor heller *ikke* for *specificitet* fra afsenderens side. Det har

derfor flere tydninger. Det bliver imidlertid specifikt i fortolkeren, **F**, hvis denne finder en ekstension for det i modellen, altså hvis dets betydningsrepræsentation slutter med at indeholde en konstant på det ubestemte syntagmes plads. Hvis derimod fortolkeren ikke kan instantiere det i modellen, er det non-specifikt.

For det *tredie* giver systemet en forklaring på, hvorfor anaforisk genoptagelse kan løse tvetydigheder omkring specificitet. Hvis nemlig et ubestemt nominalsyntagme erstattes af en konstant, gemmes den på tavlen. At der senere sker en vellykket genoptagelse med anafor viser, at der fandtes basis for en sådan på tavlen, og at det ubestemte nominalsyntagme derfor har været instantieret med en konstant, altså været interpreteret specifikt. Hvis vi derimod har en stak af virtuelle tilstande, kan konstanter kun gemmes midlertidigt, nemlig til stakken nedlægges igen. Der er altså ikke bevaret nogen information om instantiering af syntagmet, når den virtuelle modus forlades. Men sålænge **F** opererer i denne modus, er NP_anaforer mulige.

10. Evaluering af systemet

Et modelteoretisk semantisk system som det i *Blockhead* implementerede kan altså benyttes eksperimentelt til at bringe klarhed over en række komplicerede lingvistiske fænomener, syntaktiske, semantiske såvel som pragmatiske.

Metoden hertil er en konkretisering eller anskueliggørelse af et erkendelsesområde beroende på, at der drages analogier til egenskaber og relationer ved elementer i et velforstået formelt system. Denne form for modeldannelse er velkendt i naturvidenskabelige teorier.

Gyldigheden af analogierne og af den forståelse, de fører til, er en kompleks affære. Her skal peges på den særligt eksperimentelle dimension. Ved et virkelighedseksperiment manipulerer man med et begrænset udsnit af virkeligheden, for at kunne drage konklusioner om et andet, evt. blot større domæne. Ved en simulation eller et tankeeksperiment forsøger man at afbilde et symbolsystem på virkelighedsdomænet og prøver at danne billeder af andre tilstande af virkelighedsområdet ved at manipulere med symbolerne. Sådanne simuleringer kan udføres på datamat, hvis de opnår en tilstrækkelig grad af formalisering. Fordelen herved er, som man kender det fra fx meteorologiens modeller, at man meget hurtigt kan overskue langt mere komplekse domæner, relationer osv. end med andre hjælpemidler. Samtidig opnår man en maksimal sikkerhed for, at trivielle "regnefejl" ikke forfalsker resultaterne. Man har selvfølgelig ingen tilsvarende håndfast garanti for, at der ikke er fejl i forudsætningerne, som kan forfalske resultaterne.

Hvis et på datamat implementeret system reagerer i overensstemmelse med vore forventninger inden for et veldefineret testområde, plejer vi at slutte, at systemets regler er isomorfe med lovmæssigheder, der gælder for det studerede område, og at hver udvidelse af systemet, der fortsætter med at opfylde vore forventninger, udvider vor erkendelse af det område, vi slutter analogt til. Systemet fungerer som instrument i en erkendelsesproces.

Blandt de fænomener, *Blockhead* simulerer forståelse af er: bestemthed, komplekse nominalsyntagmers semantik mht. relativkonstruktioner og andre adled, anaforiske relationer i diskurser, specificitet.

Der kan heraf udledes eller verificeres en række grammatiske "love" for disse områder, der forklarer fænomener omkring forståelsen af nominalsyntagmer, deres afgrænsning, referenceforhold, anaforiske egenskaber, anvendelse i relativsætninger mv.

Det har imidlertid også interesse af vurdere mulighederne for udvidelser af systemet og den ny erkendelse, denne forventelig kan kaste af sig.

To mulige udvidelser af dette system forekommer særligt spændende: udvidelse til at omfatte forståelse af *pluralis* og udvidelse til at omfatte forståelse af *berettende tekst*.

Vedrørende *pluralis* er der to veje at gå. Man kan forsøge at behandle nominalsyntagmer i pluralis som udtryk, der refererer til *mængder* af kardinalitet større end 1. En anden løsning, som især er fristende, fordi den umiddelbart tillader at benytte de samme regler i ental og flertal, dvs. at kalde nøjagtigt de samme moduler, kunne bestå i at behandle pluralis *iterativt*. Hermed menes, at udsagn af formen

(16) a. do something with /2/3/.../some/.../all/...entities!
b. do /2/3/.../any/.../all/...entities do something?

behandles som *2, 3, ... , "random", ..., ekshaustive* iterationer over

(17) a. do something with some entity!
b. does any entity do something?

Herved opstår ikke mindst spændende problemer omkring behandlingen af generaliseret kvantificering:

(18) a. every block which is supported by the white block is placed on that block.
b. the block which is supported by the white block is placed on that block.

(18) kan ikke vedrøre en eller flere specifikke klodser. Det er fx sandt uanset om *the block which is supported by the white block* har en referent eller ej. Derfor kan generaliseret flertal ikke bero på iteration, men må bero på en logisk slutningsproces, der evaluerer udsagnet som en tautologi. Dermed opstår behovet for en eller anden form for inferens-modul i fortolkeren.

Vedrørende *berettende tekst* er man i den situation at måtte tage stilling til den videre betydning af begrebet diskursunivers. Som dette optræder i *Blockhead,* er det binært og omfatter dels entiteterne i en database, som man kunne hævde, at der refereres *deiktisk* til, dels NP_anaforerne, som der refereres *anaforisk* til. Der er ikke mulighed for i *Blockhead* at introducere entiteter, hvis eneste "ontologiske basis" er, at de omtales i diskursen. Men det er klart, at beretninger om ikke umiddelbart tilstedeværende fænomener er en så vigtig del af vor sprogbrug, og at det meste af vor viden er baseret på sådanne beretninger: reportage, rapportering, faglitteratur, fiktion osv., at den må med i eksperimentelle undersøgelser.

Der findes forskellige tilløb til løsning af disse problemer i form af såkaldte diskursrepræsentationsteorier. Vanskeligheden ved at give disse teorier tilfredsstillende implementeringer er de mangelfulde semantiske repræsentationsformalismer, der ligger til grund og som oftest bygger på den rene første ordens prædikatskalkyle. Et andet problem rejses med, at beretningers forståelighed ofte beror på encyklopædisk viden, fx om relationer mellem helhed og dele *(the underside of the white block),* ejerforhold *(the owner of the pyramid).* Her overskrides imidlertid ofte grænsen mellem lingvistisk eksperiment på veldefineret grund og forsøg på simulering i stor skala.

11. Kort bibliografi

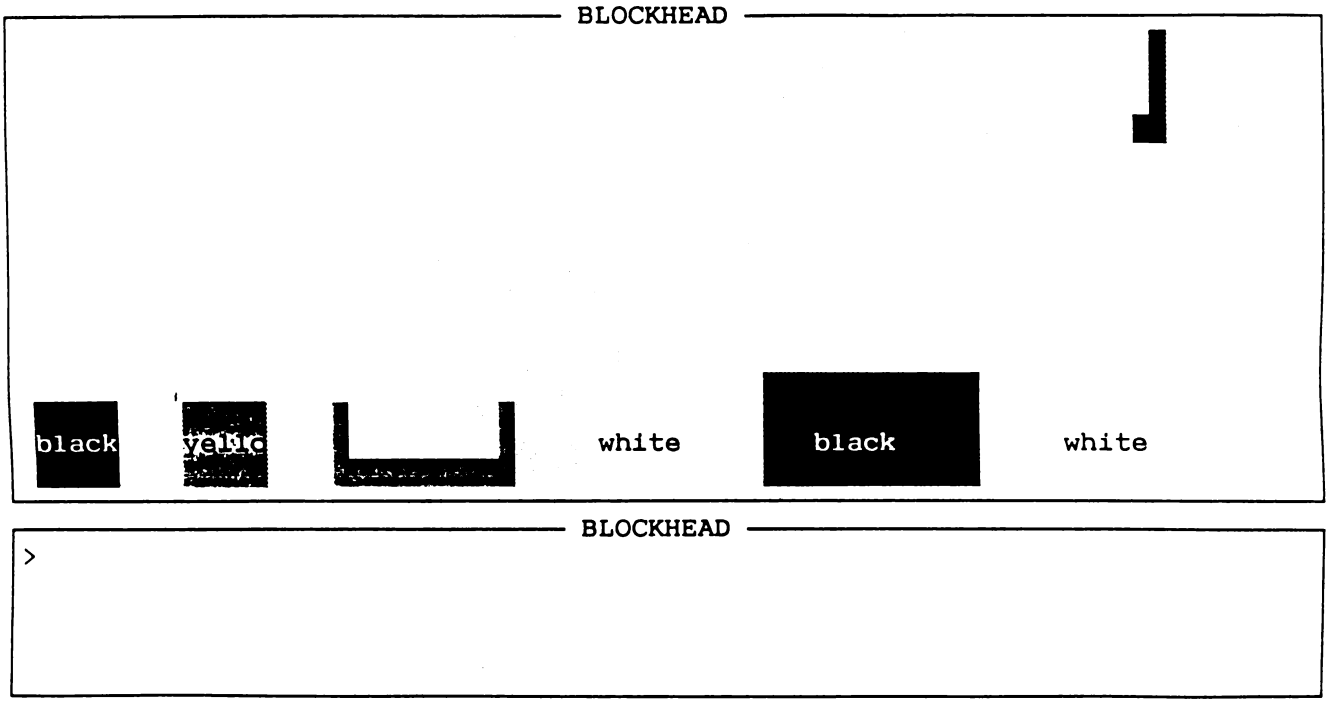
Om modelteori:

J. Allwood, L. Andersson, Ø. Dahl: *Logic in Linguistics,* 1972.
J. Barwise (ed): *Handbook of Mathematical Logic,* 1978.
B. Maegaard, H. Prebensen, C. Vikner: *Matematik og lingvistik,* 1975, kap II.

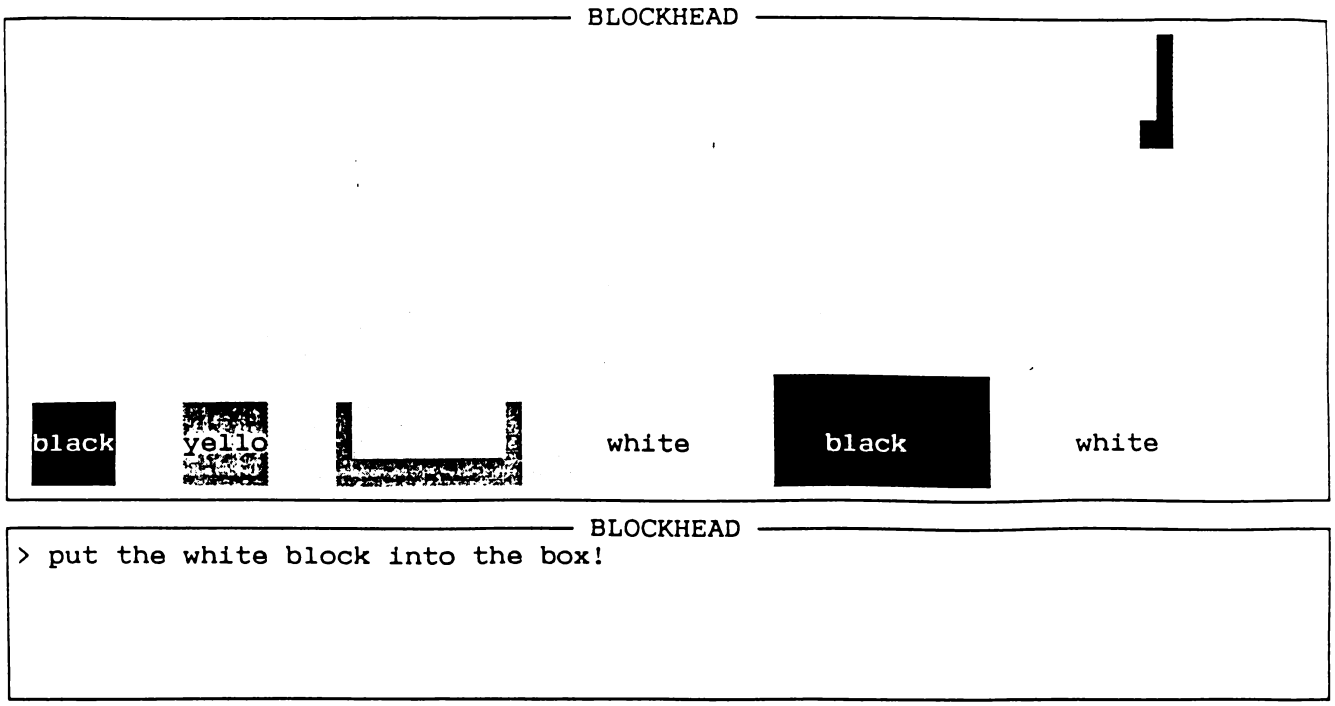
Om anaforer og Blockhead:

Data Humana 4. Humanistisk edb-center. 1987. (Med yderligere bibliografi).
Turbo Prolog. Advanced Guide. Kap. 8 (udkommer i 1988).

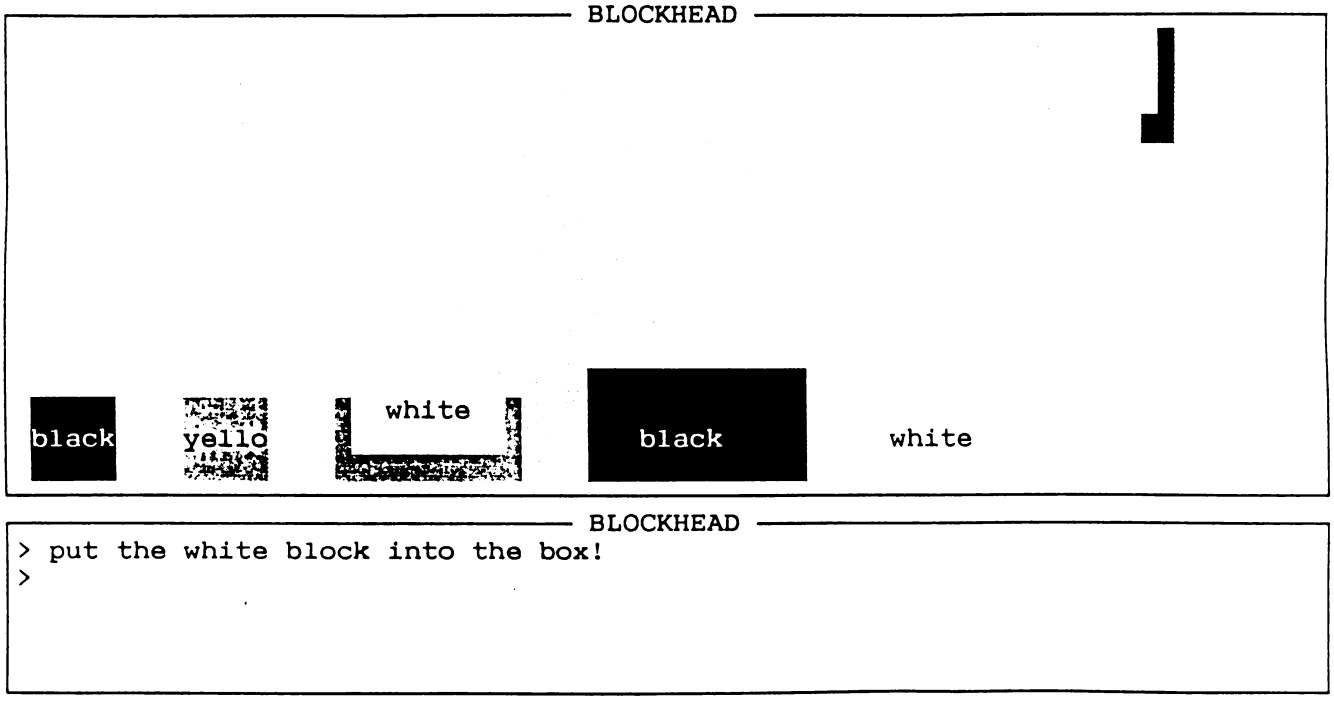
12. Appendiks: Eksempler fra Blockhead-dialog



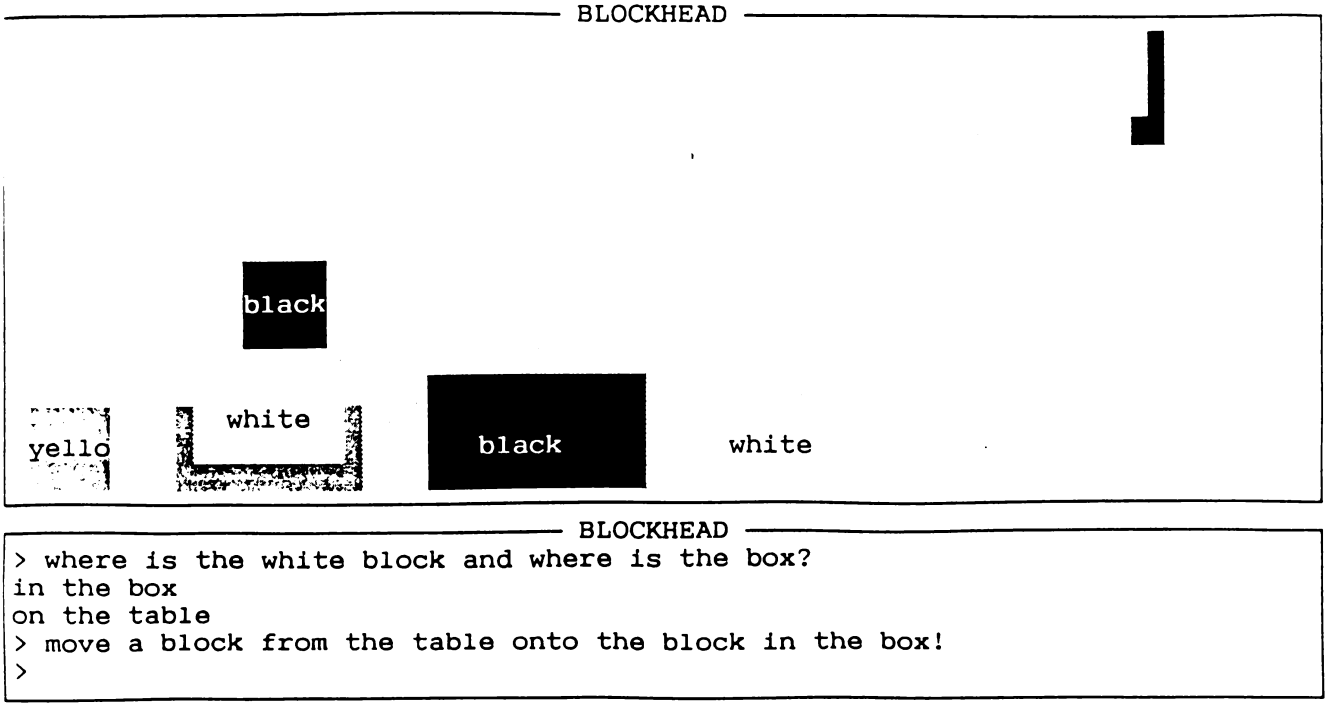
Figur 1: *Blockheads* udgangsposition



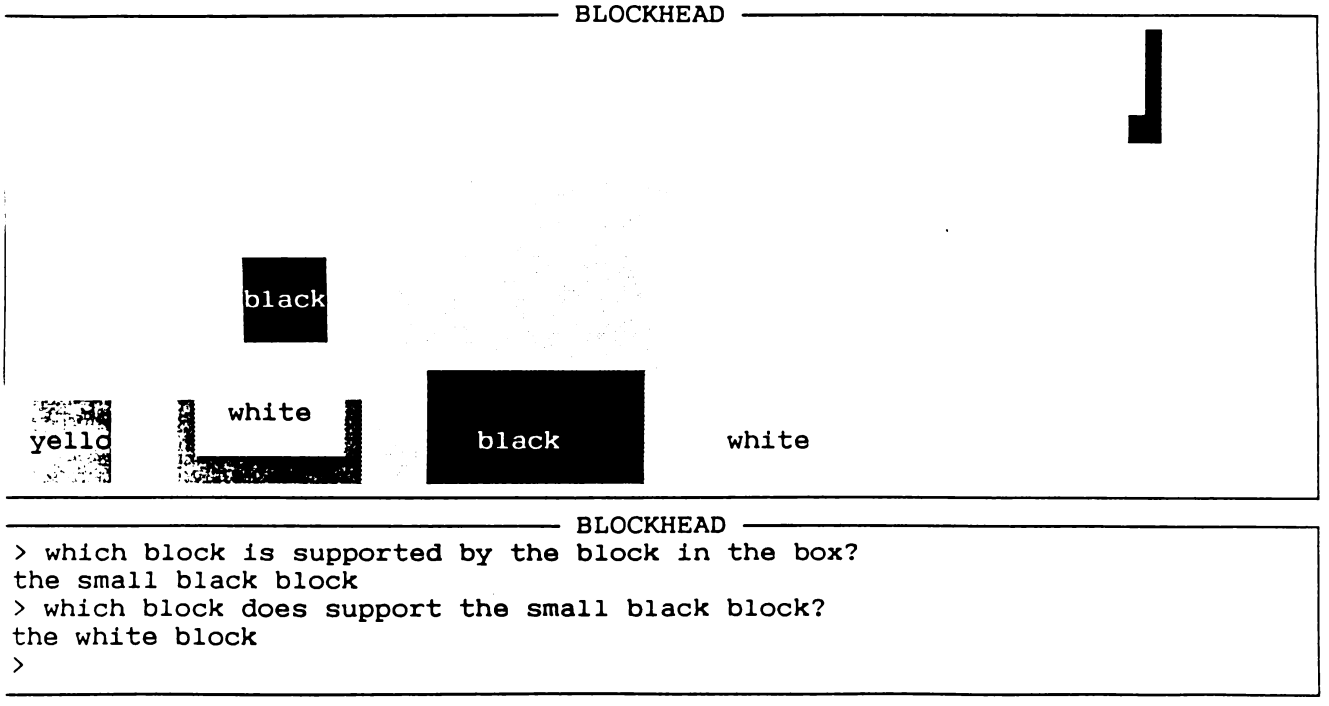
Figur 2: *Blockhead* modtager en en ordre



Figur 3: Ordren udført



Figur 4: *Blockhead* besvarer et spørgsmål



Figur 5.

Lars Ahrenberg

Department of Computer and Information Science
Linköping university
S-581 83 LINKÖPING
SWEDEN

A system for object-oriented dialogue in Swedish

1. Introduction

Two models for semantic interpretation that are currently being developed are constraint-based models (e.g. Fenstad et al. 1985, Halvorsen 1987) and models employing object-oriented knowledge representation formalisms such as frame systems or semantic networks (e.g. Bobrow&Webber, 1980; Sondheimer et al. 1984, Hirst 1987). This paper describes a dialogue system for Swedish in which I wish to combine features of both models. A large part of its linguistic knowledge, including semantic and pragmatic knowledge, is expressed as constraints. The semantic objects associated with linguistic expressions in the interpretation process are elements of a semantic network. Moreover, constraints and object descriptions play a major role also in the treatment of context.

The system, called FALIN, is being developed with the following purposes in mind: First, I want to investigate and demonstrate the possibilities of integrating syntactic, semantic and pragmatic knowledge in the interpretation process while still having that knowledge in separate modules. Second, I want to investigate the possibilities of treating dialogue phenomena such as indexicality and coherence within such a system. The results will be used in the design of a larger and more general system, LINLIN (the Linköping Natutral Language Interface; see Ahrenberg et al., 1986; Ahrenberg 1987).

As application I have chosen a simple drawing system where the human partner can draw, manipulate and ask questions about geometrical figures on a screen. The reason for this choice is that a visible domain makes it quite obvious whether the system is interpreting inputs correctly or not.

The system is still under construction. The morphological and syntactic components are in operation while the semantic components are still to be integrated in the system and the pragmatic components do not yet exist. In this paper I therefore concentrate on the problem of expressing and distributing semantic constraints, i.e. the rules that express the contributions of lexical and grammatical elements to the interpretation of the expressions of which they are part. First, I give a short overview of the system's architecture.

2. System overview

The interaction with FALIN is restricted to simple sequences of the kind that can be expressed by finite automata. The basic sequences are, with the user's moves first: Question/Answer, Instruction/Execution and Assertion/Acceptance. The system may also ask questions of the user in the process of interpretation and inform him/her of problems with the input.

The system will always try to classify an input in terms of the illocutionary categories that are allowed. This classification to a large extent determines what actions the system will execute and what information it will present to the user.

The analyzer and the knowledge bases that it has access to are illustrated in figure 1.

The *morph dictionary* consists of a stem dictionary and a set of affix dictionaries, all of them compiled into letter trees. All entries are in their surface form (cf. Karlsson, 1986). Fixed expressions comprising more than one graphical word such as *i dag*

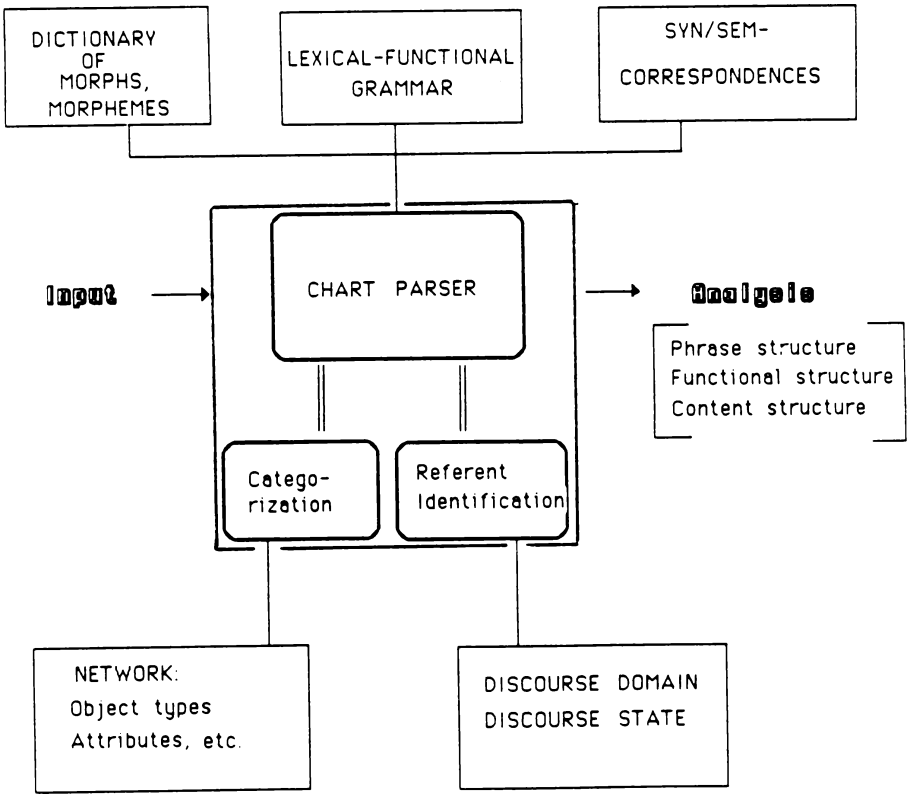


Figure 1: An overview of FALIN's analyzer.

(today) or *hur många* (how many) are included in the stem dictionary. The morph dictionary can be searched in different modes, e.g. one may choose to look for only one analysis of a given string, or all of them, or one may include or exclude the possibility of analyzing a word as a compound.

A morph in the dictionary is associated with a set of morphemes. With each morpheme there are associated a continuation class of suffix lexicons and, optionally, a flag guiding the continued search. A morpheme is either a stem or an affix. A stem morpheme carries information about syntactic category, morphosyntactic features and meaning. The meanings of a stem morpheme are collected in a *lexeme set*, where a lexeme identifies a unique semantic object as value of a semantic attribute. Basically, there is one lexeme for each sense of the morpheme. An affix morpheme is associated with morphosyntactic features and, possibly, information about category changes that it induces.

Given a string such as *cirklarna* (the circles) the dictionary search will result in the structure (1a). The first element of this structure, N, indicates syntactic category and the second element, !Cirkel, identifies a lexeme set. The content of the lexeme set may be (1b) where each different item identifies a node in the network. At that node further information about this sense of the morpheme can be found. For instance, &Circle#1 may represent the geometrical concept of a circle whereas &Circle#2 may represent the sense of "study circle".

(1a) (N (!Cirkel) ((GENDER Utral)
(NUMBER Plural)
(SPEC Definite)
(CASE Unmarked))))

(1b) !Cirkel = ((TYPE &Circle#1) (TYPE &Circle#2))

The *Lexical-Functional Grammar* is a phrase-structure grammar with annotated functional schemata in the style of Kaplan&Bresnan (1982). It deviates in several respects from the current theory and practice of LFG, however. There are no semantic forms and no attribute PRED. Instead of PRED an attribute LEX is used. The value of LEX is a lexeme set. An important difference between LEX and PRED is that LEX is not obligatory. Consequently properties such as coherence and completeness of functional structures are not determined by functional information, but are induced from semantic constraints associated with object type definitions.

In the interpretation process an input sentence is assigned three structures: a constituent structure (c-structure), a functional structure (f-structure) and a semantic

structure (s-structure). The c-structure is a phrase-structure tree whereas the other two structurus are descriptor structures encoding information in terms of attributes and values. The f-structure encodes grammatical information, in particular information about grammatical relations and morphosyntactic features. The s-structure encodes information about the input sentence regarded as a message. Thus, it is not a semantic structure in a strict sense, since it represents a contextually adequate interpretation of the input and contextual factors are used in its construction. Partial structures for sentence (2) are shown in figures 2a-2c.

(2) Rita en cirkel i övre högra hörnet.
(Draw a circle in the upper right corner.)

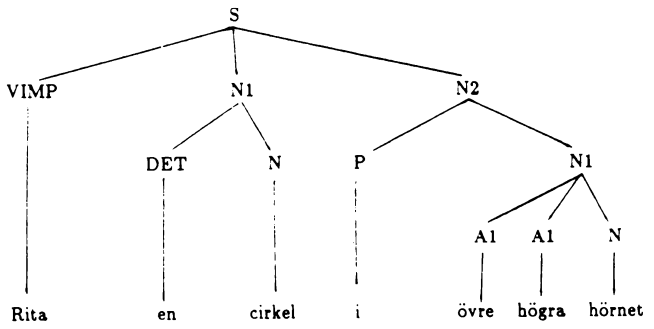


Figure 2a: *A constituent structure.*

LEX	!Rita	
MOOD	Imperative	
VFORM	Imperative	
OBJ1	LEX	!Cirkel
	NUMBER	Sing
	SPEC	Indef
	CASE	Unmarked
	GENDER	Utral
POBJ	LEX	!Hörn
	PCASE	!I
	NUMBER	Sing
	SPEC	Def
	GENDER	Neutral
	CASE	Unmarked
	MOD	LEX !Över
		SPEC Def
	MOD	LEX !Höger
		SPEC Def

Figure 2b: *A functional structure.*

TYPE	&Instruction	
CARD	1	
MODE	Constituted	
SPEAKER	<DS SPEAKER>	
ADDR	<DS ADDRESSEE>	
ACT	TYPE	&Draw
	CARD	1
	MODE	Not-given
	AGENT	<ADDR>
	RESULT	TYPE &Circle
		CARD 1
		MODE Not-given
	LOC	TYPE &Corner
		CARD 1
		MODE Given
		VPOS Up
		HPOS Right

Figure 2c: *A semantic structure.*

To be well-formed the three structures must be in a relation of *proper correspondence*. The constraints on proper correspondences between c-structure and f-structure are stated in the lexical-functional grammar whereas the constraints on proper correspondences between f-structure and s-structure are included in the definitions of individual object types and attributes. Also functional attributes are assigned such constraints. I refer to these latter rules collectively as Syntactic/Semantic correspondences, or Syn/Sem-correspondences for short.

The domain knowledge of the system is encoded in a semantic network with data structures representing object types, object instances and attributes. The object types represent concepts such as "circle", "line" and "instruction" and carry information about supertypes and subtypes, part-whole relationships and "prototypes". A prototype expresses constraints on the values of attributes that are allowed for instances of the type. As said above they also carry linguistic information specific to the type. For instance, the object type for "circle" will contain the information that it is included in the lexeme set !Cirkel. The object type for "instruction" will contain the information that an instruction can be constituted by means of an imperative utterance. Similarly, attributes representing semantic roles contain information about how they are expressed linguistically, whether by lexemes or grammatical relations.

An object instance has a unique internal name and a description. An illustration is given in (3).

(3) Circle29:
```
((TYPE       &Circle#1)
 (CENTRE     Point13)
 (RADIUS     6)
 (COLOUR     Black)
 (RESULT-OF  Draw4))
```

The discourse domain basically consists of all the objects that exist, i.e. are part of the network at any given stage in the discourse. However, without imposing some kind of stratification on the discourse domain it will not be possible to handle anaphoric or implicit reference. There have been various suggestions how this should be done (e.g. Grosz, 1977; Alshawi, 1987). The first method that will be explored in this system is to introduce an object representing the system's view of "a dialogue state" at any given moment. The description of this object, which will comprise context factors such as speaker, addressee, current topics, current visible objects etc, will then be updated for each new utterance.

The processor consists of a chart parser communicating with modules that classify descriptions and determine their referents, if any. The chart parser presently works in a bottom-up mode building c-structure and f-structure in parallel. Thus, the consistency of functional information is checked whenever a task is executed. The

parser has certain deterministic traits, which I will not describe here, but it will always find an analysis if there is one.

The role of the classifying component is to determine an appropriate object type for an s-structure constituent. Sometimes a TYPE-descriptor can be determined easily from the lexical information, but there are several complications, such as disambiguation and the handling of headless phrases. A general requirement is that, if a lexeme set has been indicated, the value of TYPE must be an element of that set. Other descriptors of the semantic structure are required to be compatible with the TYPE-descriptor according to its prototype.

The task of the referent identification component is to determine referents of the description found in an s-structure constituent. Not all s-structure constituents will refer to an already existing individual, of course. For these there is still a need to determine a mode of application of the description, i.e. the conditions under which a referent will exist.

The semantic structure associated with a constituent will normally not be constructed until the constituent is judged syntactically complete by the parser, i.e. when an inactive edge is proposed for introduction into the chart. Thus, a constituent such as *en svart fråga* (a black question) may be rejected by the analyzer on the grounds that descriptions of questions cannot contain descriptors refering to colour. Similarly, sentences such as (4) and (5) will be disambiguated when semantic constraints are taken into account. For instance, an active edge spanning the words *flytta cirkeln* of (5) and looking for a locative adverbial can combine syntactically with an inactive edge spanning the words *i hörnet*, but the proposed edge will be rejected on semantic grounds, since the location expressed by the latter words won't be of the appropriate type for a movement action.

(4) Rita cirkeln i hörnet.
(Draw the circle in the corner.)

(5) Flytta cirkeln i hörnet.
(Move the circle in the corner.)

3. Rules for syntactic/semantic correspondences

The relation between syntactic structure and semantic structure is perceived in different ways by different theories. Often some form of an isomorphism hypothesis is adopted. In formal semantics and other schools adopting a "rule-to-rule"-principle the correspondence is a derivational correspondence, not a structural one. This approach has also been used in natural language processors, e.g. in the Rosetta project (Appelo

et al. 1987). Other natural language processors rely implicitly or explicitly on structural isomorphy between syntactic and semantic structures (e.g. Lytinen, 1987; Danieli et al., 1987). While I believe that simple one-to-one relations between syntactic and semantic elements are sufficient too handle simple language fragments, I also feel that there are limits to such a methodology. There are syntactic constituents that correspond to no semantic object (e.g. formal subjects and objects), there are those that correspond to more than one semantic object (e.g. locutionary and illocutionary contents) and there are cases where several syntactic constituents relate to one and the same semantic object (e.g. idioms, adjectival attributes). Such structural modifications are easily expressed by descriptor schemata. Moreover, semantic schemata can be associated with syntactic objects and, in the other direction, functional schemata can be associated with semantic objects. Also, descriptor schemata can be associated with contextual factors in very much the same way as they are associated with syntactic objects.

Another question is what syntactic constituents should be considered relevant for the correspondence rules. Halvorsen (1983) defines the correspondences in terms of translation rules which associate functional structures with semantic structures. The semantic structures have quite a restricted form, however, (equivalent to formulas of illocutionary logic) and employ only a limited number of attributes.

Halvorsen (1987), on the other hand, states the correspondences already at c-structure level. The correspondences between functional and semantic structures are captured by means of a projection operator, σ. The projection operator takes functional structures as arguments and returns the corresponding semantic structure. A schema associating the subject constituent with the first argument of a verb is written as in (6).

(6) $((\sigma \uparrow)\ \text{ARG1}) = (\sigma(\uparrow\ \text{SUBJ}))$

Schemas of this kind are attached both to lexical entries and to rules in the grammar. A schema such as (6) would be attached to every verbal stem in the language that allows this correspondence, i.e. the great majority of verbs. The lexical entry for the verbal stem ***kick*** is specified as follows (*ibid.* p. 9):

(7) KICK V S-ED
$((\sigma \uparrow)\ \text{REL}) = \text{KICK}$
$(\uparrow\ \text{PRED}) = \text{'KICK'}$
$((\sigma \uparrow)\ \text{ARG1}) = (\sigma(\uparrow\ \text{SUBJ}))$
$((\sigma \uparrow)\ \text{ARG2}) = (\sigma(\uparrow\ \text{OBJ}))$

There are some disadvantages with this method, however. First, correspondences of the type in (6) are not stated as rules, in particular not as rules about subjects and first arguments, but as specific information about individual words, and, since there are many alternative correspondences, lexical entries tend to be overloaded with information. This is actually a general problem with lexical-functional grammars where

lexical entries are fully specified. Second, the role of the functional predicate 'KICK' is unclear. If information about predicate-argument structure is moved from functional structure to semantic structure, as Halvorsen suggests it should, it seems to be of very little significance.

In FALIN correspondences of the type (6), although in a slightly different form, are associated directly with the attributes SUBJ and ARG1 as elements of the network. Through inheritance they become available to any relation that accept ARG1 (or one of its subattributes) as an attribute.

Semantic attributes such as ARG1 and ARG2 can be regarded as abstract semantic roles (cf. Wachtel 1987). Roles such as being the agent of an act of drawing or the speaker of an utterance are differentiations of ARG1, whereas the result of a drawing, i.e. the picture, and the message of an utterance are differentiations of ARG2. Although these attributes are not in themselves representing grammatical functions, they allow the formulation of simple rules for the interpretation of grammatical relations.

Rules that induce a different mapping between grammatical relations and semantic arguments, such as rules for passive constructions, will also have their results stated on the descriptions of the attributes involved instead on the descriptions of individual verbs. Individual verbs need only be specified for the kinds of mapping they permit. Thus, if we include both the active and the passive cases in the same rule, we get something of the form of (8). The arrows have their usual interpretations as metavariables for corresponding structures. To distinguish functional and semantic structures the latter are indexed by a lowered 's' and the former by an 'f'. Schemas without arrows state conditions on the structure in which the attribute itself occurs.

(8) SUBJ: $\{$ (PASSIVE YES) $(\uparrow_s \text{ ARG2}) = \downarrow_s$ /
(PASSIVE NO) $(\uparrow_s \text{ ARG1}) = \downarrow_s \}$

Conversely, the description of ARG1 will be as in (9), where (AV OBJ) identifies the agent relation in a passive clause.

(9) ARG1: $\{ (\uparrow_f \text{ PASSIVE YES}) \ (\uparrow_f \text{ AV OBJ}) = \downarrow_f$ /
$(\uparrow_f \text{ PASSIVE NO}) \ (\uparrow_f \text{ SUBJ}) = \downarrow_f \}$

By distributing the functional schemas in the semantic network we reduce much of the lexical overloading in ordinary lexical-functional grammars. Every different sense of a morpheme is given its own entry. Moreover, when a stem is part of an idiom or other polymorphemic item, information about this is not only attached to the stem, but also

to the relevant node in the network. For instance, the morpheme *ta* (take) is associated with a LEX-value, !Take, that have a fairly large number of different senses. In this set we would also find the action &Take-away, expressed in Swedish as *ta bort*. This item is distinguished from all the others in the same set by a special condition on functional structures expressing it, i.e. that it contains the two descriptors in (10) at top level. Here, PRT is an attribute representing a verbal particle.

(10) &Take-away

$$(\uparrow_f \text{LEX}) = \text{!Ta}$$
$$(\uparrow_f \text{PRT LEX}) = \text{!Bort}$$

A functional structure may correspond to a content structure in two different modes. I distinguish a ***constitutive*** (or ***illocutionary***) mode from a ***strict*** (or ***locutionary***) mode. The utterance of an expression constitutes an illocutionary act, i.e. an object instance of a particular illocutionary type. The description of this object is said to correspond to the functional structure of the expression in the constitutive mode. The descriptions of the objects referred to in the utterance, on the other hand, are said to correspond strictly with the f-structures of their refering expressions. Constitutive correspondence will be indicated by double arrows, $\Uparrow$ and $\Downarrow$, to distinguish it from strict correspondence.

Of the linguistic elements that participate in constitutive schemata I will here only consider mood descriptors. A rule for the imperative mood may be formulated as follows:

(11) (MOOD Imperative):

$$(\Uparrow_s \text{TYPE}) = \text{\&Instruction}$$
$$(\Uparrow_s \text{AGENT}) = \text{<DS SPEAKER>}$$
$$(\Uparrow_s \text{PATIENT}) = \text{<DS ADDRESSEE>}$$
$$(\Uparrow_s \text{ACT}) = \uparrow_s$$
$$(\Uparrow_s \text{ACT ARG1}) = (\Uparrow_s \text{PATIENT})$$

Here DS is a reference to the description of the discourse state. When an s-structure is constructed by means of (11) the current values for the indicated attributes of the discourse state will be retrieved. The fourth schema relates the two different corresponding s-structures to each other, thus integrating the locutionary meaning into the description of the illocutionary act.

To be properly corresponding an f-structure and an s-structure must meet certain general requirements. The functional attributes and descriptors can be divided into two classes, semantically relevant and semantically irrelevant. The latter descriptors

play no role in the correspondence relation, whereas every semantically relevant functional descriptor must correspond to a structure of semantic descriptors according to one of the syn/sem-correspondences defined for it. Both f-structures and s-structures must be consistent and determined. Moreover, the s-structure constituents must be typed, compatible with a prototype and specified as to how they apply as descriptions of objects in the discourse domain. Not all information in s-structures have a counterpart in functional descriptors, however. It may instead be retrieved from the discourse state. All this means that there is no requirement on strict isomorphy, whether derivational or structural, between f-structures and s-structures. Still, the use of schemata and the postulation of only two classes of correspondences make the framework both principled and restricted.

Acknowledgements

This paper reports work in progress of the project "Analysis and Generation of Natural Language Texts" financed by the National Swedish Board for Technichal Development. I am indebted to Nils Dahlbäck, Arne Jönsson, Magnus Merkel and Mats Wirén for valuable discussion and to Ulf Dahlén for much of the programming.

References

Ahrenberg, L. Dahlbäck, N., Jönsson, A., Merkel, M. och Wirén, M. 1986. Mot ett dialogsystem för svenska. *NLPLAB Memo 86-01*. Department of computer and information science, Linköping university: Linköping.

Ahrenberg, L. 1987. Parsing into Discourse Object Descriptions. In *Proceedings, Third Conference of the ACL European Chapter*, Copenhagen 1-3 April, 1987, pp. 140-147.

Alshawi H. 1987. *Memory and context for language interpretation.* Cambridge University Press: Cambridge.

Appelo, L., Fellinger, C. and Landsbergen, J. 1987. Subgrammars, Rule Classes and Control in the Rosetta Translation System. In *Proceedings, Third Conference of the ACL European Chapter*, Copenhagen 1-3 April, 1987, pp. 118-133.

Bobrow, R. J., Webber, B. L. 1980. Knowledge Representation for Syntactic/Semantic Processing. In *Proceedings, First Annual National Conference on Artificial Intelligence*, Stanford, August 1980, pp. 316-323.

Fenstad, J. E., Halvorsen, P-K, Langholm, T. and van Benthem, J. 1985. Equations, Schemata and Situations: A framework for linguistic semantics. Manuscript, CSLI, Stanford University.

Grosz, B. J. 1977. The Representation and Use of Focus in Dialogue Understanding. (PhD thesis) *SRI Technichal Note No. 151*, SRI International: Menlo Park.

Halvorsen, P-K. 1983. Semantics for Lexical-Functional Grammar. *Linguistic Inquiry 14:4*, 567-615.

Halvorsen, P-K. 1987. Situation Semantics and Semantic Interpretation in Constraint-based Grammars. *Technichal Report CSLI-TR-87-101*, Centre for the Study of Language and Information: Stanford.

Kaplan, R. & Bresnan, J. 1982. Lexical-Functional Grammar: A Formal System for Grammatical Representation. In Bresnan J. (ed.) 1982: *The Mental Representation of Grammatical Relations*, The MIT Press: Cambridge Mass., pp. 173-281.

Karlsson, F. 1986. A paradigm-based morphological analyzer. In Karlsson, F. (ed.) 1986: *Papers from the Fifth Scandinavian Conference of Computational Linguistics, Helsinki, December 11-12 1985.* University of Helsinki: Helsinki, pp. 95-112.

Lytinen, S. L. 1987. Integrating syntax and semantics. In Nirenburg, S. (ed.) 1987: *Machine translation.* Cambridge University Press: Cambridge, pp. 302-316.

Moreno, D., Ferrara, F., Gemello, R. and Rullent, C. 1987. Integrating Semantics and Flexible Syntax by Exploiting Isomorphism between Grammatical and Semantical Relations. In *Proceedings, Third European Chapter ACL Conference*, Copenhagen, April 1-3, 1987, pp. 278-283.

Sondheimer, N. K., Weischedel, R. M. and Bobrow, R. J. 1984. Semantic Interpretation Using KL-ONE. In *Proceedings of Coling '84*, Stanford University, Cal. 2-6 July 1984, pp. 101-107.

Wachtel, T. 1987. Discourse structure in LOQUI. In *Recent Developments and Applications of Natural Language Understanding.* UNICOM Seminars Ltd: London, pp. 161-86.

Lars Borin
Uppsala University
Center for Computational Linguistics
Box 513
S-751 20 Uppsala, Sweden

A constraint-based approach to morphological analysis (preliminaries)

0. Introduction

This paper is intended to give the background to ongoing work on a constraint-based system for morphological analysis, which I intend to present in more detail later. The system represents one stage in the development of a (computational) morphological formalism suited for modelling the mechanisms of word formation, a development that began about four years ago, when I started experimenting with Koskenniemi's (1983) two-level model (Borin 1985; 1986a; 1986b). A search of the relevant literature in both linguistics and computational linguistics showed remarkable similarities in many newer approaches to language, similarities that in some ways represent a return to an older linguistic tradition. These approaches can all be said to advocate *relational* models of language, a notion that will be discussed below.

The particular approach proposed here was directly inspired by a somewhat older linguistic model, but still a relational one, namely stratificational grammar and its offshoots (see e.g. Gleason 1964; Lamb 1966; Lockwood 1972; Reich 1969; 1970), to my knowledge the most thorough attempt to formalize the structuralist notion of language as a system where everything depends on everything else, i.e. a system of relations.

In section 1 below I try to give a characterization of relational linguistic models and also to give an overview of some of the relational models found in the literature. Section 2 discusses the possibilities of using a recent artificial intelligence technique, constraint systems (Hein 1981; 1982; Maleki 1987), as a general implementation language for these models. Conclusions and some directions for further research are the topic of section 3.

1. Relational linguistic models

In the last few years, there has been a convergent development in the closely related areas of linguistics, computational linguistics, artificial intelligence and cognitive science towards relational models of language and language use. The models I am referring to are (at least), for linguistics: the Meaning ⩽ = ⩾ Text model of Mel'čuk and others (Mel'čuk 1974; Mel'čuk & Pertsov 1987), autosegmental phonology (Goldsmith 1976) and morphology (McCarthy 1981; 1982); for computational linguistics: two-level morphology (phonology) and other finite-state phonological and morphological models (Koskenniemi 1983; Kay 1987) and, at least partly, some of the unification-based formalisms, like Lexical-Functional Grammar (LFG) (Bresnan & Kaplan 1982), Functional Unification Grammar (FUG) (Kay 1985) and Uppsala Chart Parser (UCP) (Sågvall Hein to appear); for artificial intelligence: associative networks, also called semantic networks and conceptual graphs (see e.g. Sowa 1984 or the articles in Findler 1979); for cognitive science: connectionist models (e.g. Dell & Reich 1980; Dell 1985).

Tentatively, we may characterize relational models of language as models with the following properties:

- relations are more important than processes in the model, this in contrast to e.g. generative grammar or Hockett's (1954) IP (Item-and-Process) model. As a rule, there is only a small number of relations in the model.

- linguistic units, or items, if they have any theoretical status at all, are defined through their relations to other units;

From the point of view of computational linguistics, this has some important consequences for the way processing is considered to be carried out in a relational linguistic model:

- the model is decentralized, in the sense that there are many autonomous processing elements. On the other hand, there are only a few element types;
- the processing elements and their interconnections (links) may be seen as a *network*, where the topology of the network is an important part of the model. Just as there are only a few element types, there is only a small number of possible link types between the elements;

1.1 Some relational linguistic models

First of all, one must mention classic Saussurean structural linguistics. Here, language is viewed as a system where every unit is defined by its place in the totality, i.e. by its relations to other units:

> Units and grammatical facts would not be confused if linguistic signs were made up of something besides differences. But language being what it is, we shall find nothing simple in it regardless of our approach; everywhere and always there is the same complex equilibrium of terms that mutually condition each other. Putting it another way, *language is a form and not a substance*.
>
> (de Saussure 1959:122, emphasis in the original)

Being in Copenhagen, I should not forego to mention Hjelmslev and glossematics in this context.The notion of language as a system of relations was very much present in Hjelmslev's work (e.g. Hjelmslev 1961), and workers in stratificational grammar usually mention him as their single most important source of inspiration.

Against this background, the more recent theories represent a return to a tradition that has lived in the shadow of the preoccupation with process-based linguistic description that has characterized much of (especially American and generative) linguistics during the last three decades or more. This does not mean that history has taken a full circle; rather, the insights of the structuralists are now combined with a formal rigour that to date has been the foremost contribution to linguistics by generative grammar and computational linguistics.

1.1.1 The Meaning ≪ = ≫ Text model

The Meaning ≪ = ≫ Text model concerns itself with the relation between the two entities in its name, i.e. for a given meaning, how do you get to the (very large number of) texts that express this meaning, and conversely, how do you get to the (several possible) meanings of a particular text. To this end, the model operates with seven levels of linguistic representation: semantic, deep and surface syntactic, deep and surface morphological and deep and surface phonetic. To get from one level to the next level up or down, there are interlevel relations, many-to-many mappings between the levels. These mappings have been described in various ways: in an earlier version the model (Mel'čuk 1974) as chains of ordered transducers, while at present they are held to be a set of unordered correspondence rules, which

> are conceived of not as prescriptions, or instructions of an algorithm, but rather as permissions and prohibitions, or statements in a calculus.
>
> (Mel'čuk & Pertsov 1987:35)

Since this model was proposed by Russian linguists, the lexicon has a very important role in it. There is a carefully specified format for the lexicon to be used in the model, and some actual lexicons (Mel'čuk 1984; Mel'čuk & Žolkovskij 1984) have been prepared according to this format, which all have the common trait that they hold very much information about each lexical unit. E.g. in the Russian lexicon (Mel'čuk & Žolkovskij 1984) the entry for the word *čuvstvo* 'feeling' is 16 pages long.

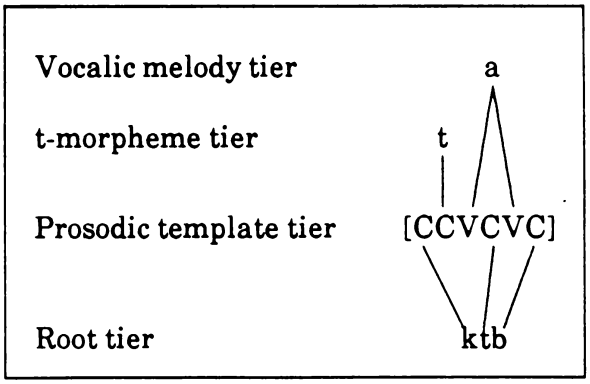
1.1.2 Autosegmental models

It has often been observed that some phonological phenomena are difficult to handle in a segmental phonology, because the phonological segment does not seem to be their right domain, but rather the syllable, the morph(eme) or the word. Among such phenomena are stress, tone (pitch), vowel harmony and synharmonism. Various solutions have been proposed for overcoming these difficulties; often stress and tone, at least, have been segmentalized into the phonemic string. Autosegmental phonology was developed as a reaction against linear, or segmental phonology, to deal with the phenomena of accent, tone etc., that present a problem to the segmental approach. The approaches most relevant in the present context, however, are prosodic analysis (Firth 1948; Robins 1957) and Harris' (1944) long and contour components. Unlike Harris' approach and like prosodic analysis, autosegmental phonology divides the traditional segmental phonemic level into a number of simultaneously occuring components. In autosegmental phonology all these components are segmental, something that sets them off from the prosodies of prosodic analysis. Furthermore, among the segmental components, or tiers, one has a special status. This is the traditional segmental tier, which serves as the coordinating tier, to which all the other - autosegmental - tiers are mapped by a general mapping relation, called the well-formedness condition (WFC). Autosegmental phonology is declared to be a further development of generative phonology, but it differs considerably from the latter in spirit:

> Autosegmental phonology is a particular claim, then, about the *geometry* of phonetic representations; it suggests that the phonetic representation is composed of a set of several simultaneous sequences of these segments, with certain elementary constraints on how the various levels of sequences can be interrelated -- or, as we shall say, "associated."
>
> (Goldsmith 1976:16, emphasis in the original)

Autosegmental phonology has been used to describe, e.g., tone and accent (Goldsmith 1976; Withgott & Halvorsen 1984) and vowel harmony (Clements 1980).

The formal devices of autosegmental phonology have also been used in morphology, notably by McCarthy (1981; 1982) for describing Classical Arabic and Hebrew morphology, i.e. strongly non-concatenative systems. McCarthy introduces some additional autosegmental tiers in the model. In his version, the material from the traditional segmental tier is distributed over several *morphemic tiers*, and the coordinating component, to which all other tiers are mapped, is the *prosodic template* or *CV-skeleton*, a (partly specified) phonotactic constraint. The general "geometry of the representation" will hopefully be illustrated by the following figure, the autosegmental representation of the Classical Arabic verb stem *ktatab* 'was registered' from the root *ktb* 'to write, writing', taken from McCarthy (1982:193):



1.1.3 Finite-state phonology and morphology

The by now well-known two-level formalism is the brainchild of Kimmo Koskenniemi of Helsinki University (see e.g. Koskenniemi 1983; Karttunen 1983), and it has given rise to a fair number of both applications to specific languages and similar formalisms,

collectively referred to as finite-state morphology. Koskenniemi is quite insistent on two-level morphology being a relational formalism:

> The two-level formalism is neutral with respect to production and analysis because it describes morphological phenomena as relations between lexical and surface representations. The relations are seen as correspondences, not as segments being transformed into other segments.
>
> (Koskenniemi 1983:10)

There are other variants of finite-state morphology that use more than two levels in the description, e.g. Kay's two-level morphology with tiers (Kay 1987), which is an implementation of autosegmental morphology using the finite-state transducers of two-level morphology for the WFC (see 1.1.2 above).

1.1.4 Stratificational grammar and relational grammar

Stratificational grammar (SG) was developed as a purely linguistic theory, just like its contemporary, generative grammar, but it was developed in a machine translation project, and its theoretical devices presumably were influenced by this fact. Stratificational grammar and its offshoot relational grammar (RG) see language as a network of relations. Some versions of SG do not give items any status whatsoever in the theory, stating that the items of linguistic description are simply nodes in the overall network. The network connects to items at its both ends, however - phonetic units at one end and conceptual units at the other. SG, but not RG, also holds that language is *stratified*, i.e. there are layers or strata of linguistic description, normally corresponding to the traditional linguistic divisions of language into phonology, morphology, syntax and semantics[1].

The relations allowed in a stratificational or relational description are usually taken from a small set of primitive relations, the most important being conjunction (symbolized by AND nodes in the network diagrams, see below) and disjunction (OR nodes). There are also the interstratal relations of realization ('is realized by'), composition ('is composed by') and their inverses ('is a realization of' and 'is a part of'). The number of relation types postulated varies among different authors, but at least these types, in some form, are present in all descriptions. The difference between the relations of realization and composition gives rise to two subsystems on each stratum, the realizational part and the tactics, where the latter acts as a filter on the realizations allowed by the former. Seeing language as a system of relations, stratificational grammar has no place for process description in the sense of Hockett's (1954) IP model, or in the interpretation of 'generate' in 'generative grammar' as meaning the same thing as 'produce' (this is not the interpretation intended originally, but common nevertheless); just like in the two-level model, all the strata are considered to exist side by side, simultaneously. An attractive trait in relational models is their inherent non-directionality (bidirectionality in this case); if you put in a text at one end of a relational network, a meaning or meanings will appear at the other end, and vice versa. Many stratificationalists have also found graphical network descriptions ("lambograms"; see section 3 below) to be a convenient tool, preferable to algebraic rule systems for the description of language.

1.1.5 Associative networks

Associative networks are perhaps more commonly known as semantic nets. The term associative network was introduced, as far as I know, by Findler (1979). Other terms that have been used for the same thing are semantic memory (Quillian 1968) and conceptual graphs (Sowa 1984). Associative network formalisms have been used in artificial intelligence both for general knowledge representation and for storing more specifically linguistic knowledge. Associative networks are directed graphs, with labelled nodes and edges. The common interpretation is that the nodes represent entities and the links relations between these entities; in a predicate calculus setting, the links would be the predicates and the nodes the arguments (constants and variables) of these predicates. In this connection it is worth noting that the highest -

the sememic - stratum in a stratificational grammar is considered by most stratificationalists to be a structure very similar to an associative network; to avoid terminological confusion, this structure is often called a *reticulum*, since the term network is reserved for the less densely connected lower strata, where temporal relations play an important role.

1.1.6 Connectionist models

Connectionist models share many of the assumptions of the models described above, but the fundamental assumption behind them is "that theories and and scientific languages based on the computational character of the brain are productive (even essential) in many areas of Cognitive Science" (Feldman 1985:1). In other words, connectionist models are based on computational architectures with many autonomous processing elements, which can be linked up in a small number of ways, and communicate with a limited repertory of (simple) signals, just like neurons. Among the other models mentioned in this section, it is perhaps stratificational grammar that has the closest affinity to connectionist models. The speech production model used by Dell & Reich (1980) is a refined variety of the relational networks earlier described by Reich (1970), and Schnelle's (1981) neurologically inspired *net linguistics* works with more or less the same element types as stratificational and relational grammar.

1.1.7 Unification-based models

Unification is a technique that has become increasingly popular in natural language processing systems. The basic data structure in unification-based grammatical formalisms is the attribute-value graph, a directed acyclic graph (DAG) made up of attributes (like *case*) which have values (e.g. *nominative*). Instead of beeing atomic, the values may, in turn, be attribute-value graphs. Unification is an operation for acertaining that two (or more) attribute-value graphs are compatible with each other, which involves checking the values of identical attributes in the two graphs. If the values are atomic, they must be identical to be compatible; if they are attribute-value graphs, unification is carried out recursively on these; if one of the values is undefined, both values become identical to the defined value. If unification succeeds, the two graphs will become identical, i.e. attribute-value pairs that appeared in only one of them will now be present in the other one as well. Unification is unordered, so in order to handle the temporally ordered surface structure of language, most unification-based formalisms have two components: a context-free phrase structure grammar which is used to build a phrase structure tree from the surface morphological and syntactic representation, annotated with functional structures. These functional structures are then unified with partly specified lexical and grammatical functional structures, sometimes called *constraining equations* (e.g. in Withgott & Halvorsen 1984), to yield a fully specified functional structure as the analysis of the linguistic unit being parsed.

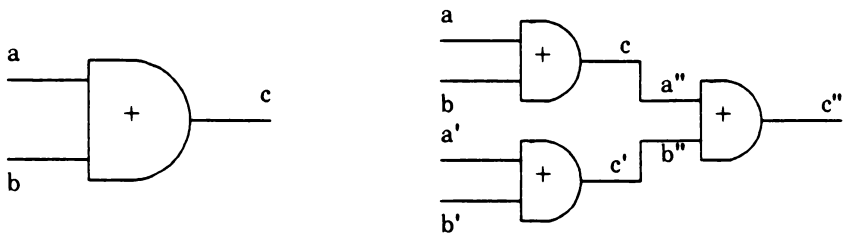
2. Constraint systems

Constraint systems is an AI technique for representing knowledge about relations among entities, values and the like. The basic building block of constraint systems is the constraint,

> an active relation between a (usually small) set of objects. The relation is termed active since it exhibits two crucial features. It establishes itself as soon as enough information about the participating objects is available and it enforces the relation once it has been established. (Hein 1981:3)

The figure below is intended to serve as an illustration of the way constraint systems are set up. The left half of the figure shows a simple constraint of the equation kind (a + constraint), which expresses the relation $a + b = c$ in the following way. As soon as the values of at least two of the three variables in the equation are known, the third variable will automatically be set to a value that satisfies the equation. After

this, the constraint will enforce the relation by reacting to changes in the values of the variables. The exact nature of the reaction, however, is dependent on the kinds of constraints in the system and their use. The right half of the figure shows that simple constraints can be connected together - via their variables - into constraint networks. The interconnections are made via equality constraints.



A simple constraint (+) and a small network of constraints

Constraint systems combine the object-oriented and declarative programming paradigms. The characteristic features of both of these are generally considered important in the perspective of computational linguistics. Object-orientedness implies decentralized control: processing control is local to a constraint. This in turn means that constraint systems would be fairly easy to implement on parallel computer architectures, like connection machines (Hillis 1984). Declarative programming, on the other hand, fits well in with the relational linguistic models discussed above. The metaphors used in talking about relational linguistic models are conspicuously close to the kinds of phenomena constraint systems are supposed to be good at handling, namely geometrical and topological relationsships (see the section on autosegmental models above), equations and (of course) constraints holding between objects and values. This makes one suspect that constraint systems should be fairly easy to use for implementing these linguistic models on a computer.

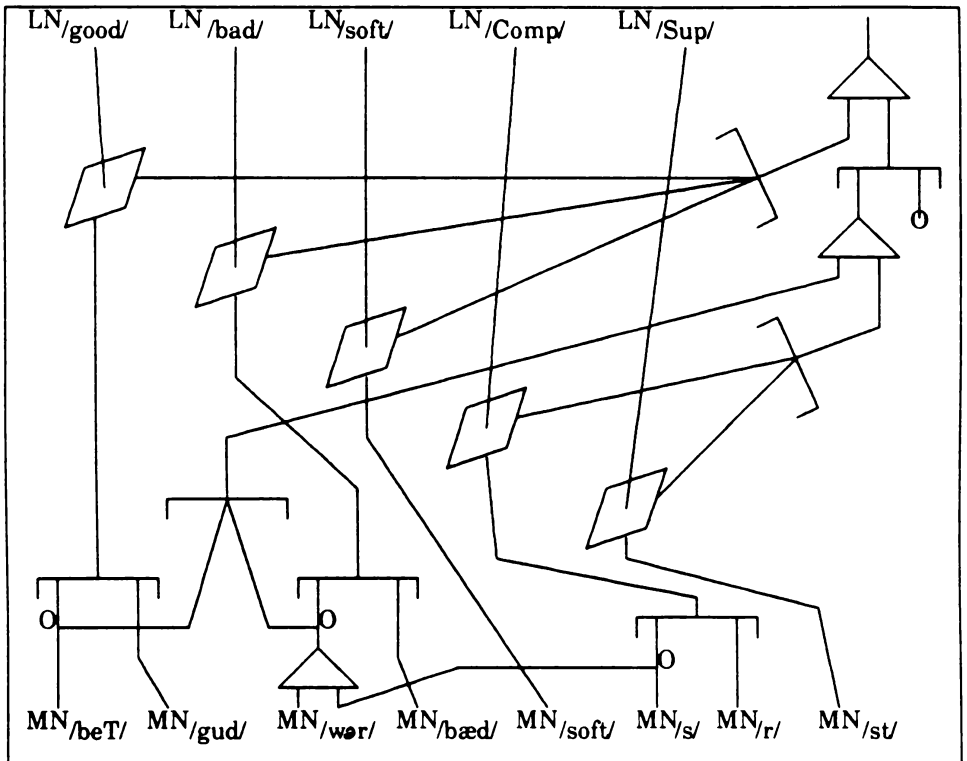
3. Conclusions and further research

Recently I have got access to an experimental constraint system implementation (ICONStraint, written by J. Maleki of Linköping University for the Xerox 11XX Lisp machines, and described in Maleki (1987)), which has made it possible for me to start experimenting with a relational model of linguistic structure, expressed as a network of constraints.

The structure of this network is heavily influenced by the networks used in presentations of stratificational grammar, mostly because this is the way of least resistance, since stratificational grammar is one of the more thoroughly formalized relational linguistic models. Its cousin, relational grammar, has been partly implemented as a computer model, Reich's (1970) *relational network simulator*. I say partly, because Reich discusses mostly language production - or encoding - while decoding is mentioned only in passing, like in most stratificational descriptions I have seen. This is one reason that I have chosen to lay the emphasis on the decoding direction in my work and consequently talk about "morphological analysis" in the title of this paper. Since more attention has been given to the encoding direction in a stratificational grammar, it seems natural to start with the structures and relations that have been shown to work in encoding and somehow reverse them to do the decoding. In the case of unrestricted rewrite rule systems, like the tree transformations used in generative grammar, this is generally not possible (King 1983), but

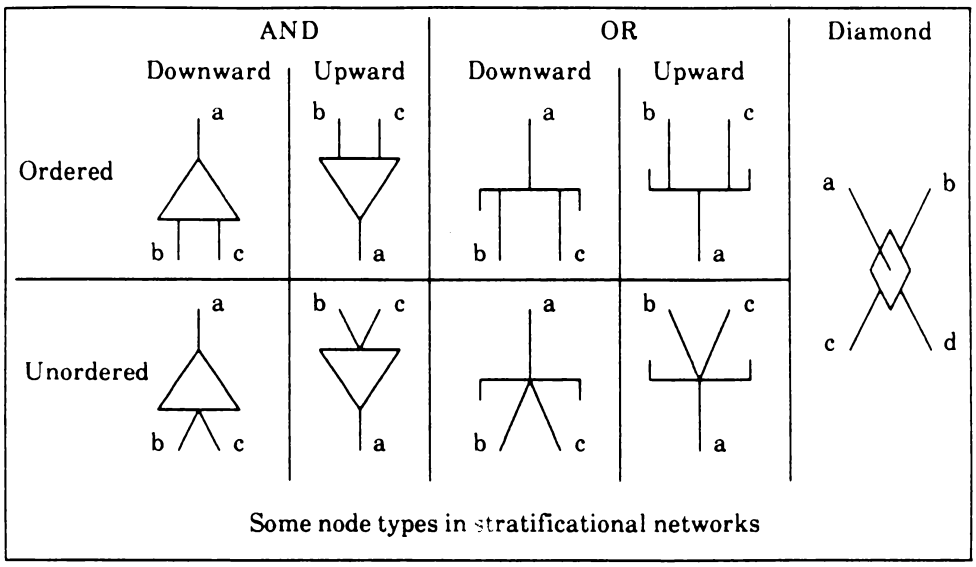
relational models should in principle be able to cope with the task. This is where constraint systems enter the picture. Being made up of active relations in the sense stated above (section 2), once a constraint network that models encoding has been built, the same network should work just as well for decoding purposes. An added requirement could be that the basic constraints in the network as closely as possible mimic the primitive relations (nodes) that are postulated in (some version of) stratificational or relational grammar, since it would then be possible to test published descriptions directly. This presupposes that these relations are well-defined in both directions.

It seems, however, that the latter is not true for the nodes in a stratificational grammar. This is partly due to the nature of the relations involved, and partly because of a certain vagueness in their definitions. I will discuss the last point first, but first I will make a small digression into the written format of stratificational descriptions. As I indicated above, the normal way of presenting such a description is in the form of one or more graphs ("lambograms"), for example the following graph that describes comparison of English adjectives:



Part of a stratificational network (Lockwood 1972:85)

The nodes always represent the same relations, while the meaning of the lines is dependent on the context; sometimes a line expresses the realization relation, sometimes it just connects one of the participants in some relation to that relation. One of the problems with the node definitions, as they are given in the literature, is that they are not detailed enough, meaning either: 1) that not all input/output combinations are accounted for, or, more often: 2) that for a given input, the output is indeterminate. The following figure shows some common node types used in stratificational networks, taken from Christie (1974).



Some node types in stratificational networks

The OR and AND nodes are discussed below. The diamond node connects the tactics and the realizational part together.

In the worst case, the nodes are given only informal definitions, like:

> Another fundamental linguistic relationship is that which a class bears to its members. In stratificational terminology, this is called an OR relationship.
> (Lockwood 1972:34)

Even if formal node definitions are given, they often are not quite formal enough (cf. Schreyer 1980; 1981); sometimes the formal definition of an individual node type leaves unclear some aspects of its function in the network as a whole. A notorious problem in this respect appears in the definition of the unordered OR, in the case of how the plural side depends on the singular side. The most informal definitions simply ignore the problem. Also, there are serious problems with timing, which are seldom or never discussed (see, however, Gleason 1980), but which become very real once an actual implementation is considered. There are two model-internal aspects to the timing problem. The first is the need of some synchronization mechanism in the model; as I said above, relational models are basically decentralized: there is no central processor that distributes processing tasks according to some internal state and clock, only many autonomous processing elements, working in parallel[2]. Since there is no central clock and since some of the nodes are described as temporal, there must be other means of synchronizing signals in the network. The other problem is in some ways dependent on the first: how do you define the relevant temporal units to use in the network and their interrelations? For example, it is said that the ordered AND describes the "important relationship in linguistic structure [...] of a combination to its [...] constituents or components" (Lockwood 1972:31), when "the order of constituents is significant" (*ibid.*). Implicit in most, if not all, definitions of the ordered AND is not only that the constituents on the plural side are temporally ordered, but also that they are temporally *contiguous*. Then it becomes important to define temporal contiguity; is it absolute, defined in terms of some minimal temporal unit, or is it relative to some events in the network? Another notion that is in need of a definition before the relations can be modelled, is that of *simultaneity*; the unordered AND node is described as one where the constituents appear "simultaneously or in no specified order" (Lockwood 1972:33). Also, constraint systems have been used mostly to represent atemporal relations, due to the nature of the problems they have been used to solve. This must not necessarily be the case; the author of the ICONStraint system has indicated the need for modelling time and change within the constraints paradigm (Maleki 1987:81).

These are the two problems that I am concentrating on at the moment: the formal node definitions and the introduction of time into the implementation language.

Acknowledgement

The research described in this paper would have been much more difficult to carry out without the ICONStraint programming system developed by Jalal Maleki of Linköping University, who kindly put the system at my disposal.

Notes

[1]For another relational theory of language, where language is stratified, and the important relations "realized by" and "composed of" are kept more consistently apart than in stratificational grammar, see e.g. Sgall *et al.* (1969).

[2]Nothing changes, in principle, even if the actual implementation, like the one described here, is made on a serial machine with one central processor, i.e. the parallelism is simulated.

References

Borin, L. 1985. *Tvånivåmorfologi: Introduktion och användarhandledning*. UCDL-L-3. Uppsala University, Center for Computational Lingustics.

Borin, L. 1986a. What is a lexical representation? Karlsson, F. (ed.): *Papers from the Fifth Scandinavian Conference of Computational Linguistics*. Publications, No. 15. University of Helsinki, Department of General Lingustics.

Borin, L. 1986b. *Swedish Two-level Morphology: Some Remarks*. UCDL-R-86-1. Uppsala University, Center for Computational Lingustics.

Bresnan, J. & R.M. Kaplan 1982. Lexical-functional grammar: a formal system for grammatical representation. Bresnan, J. (ed.): *The Mental Representation of Grammatical Relations*. Cambridge, Massachusetts - London.

Christie, W.M. Jr. 1974. A stratificational view of linguistic change. Anderson, J.M. & C. Jones (eds.): *Historical Linguistics I*. Amsterdam.

Clements, G.N. 1980. *Vowel Harmony in Nonlinear Generative Phonology: An Autosegmental Model*. Indiana University Linguistics Club. Bloomington, Indiana.

Dell, G.S. 1985. Positive feedback in hierarchical connectionist models: applications to language production. *Cognitive Science*, Vol. 9, No. 1.

Dell, G.S. & P. Reich 1980. Toward a unified model of slips of the tongue. Fromkin, V.A. (ed.): *Errors in Linguistic Performance*. New York.

Feldman, J. 1985. Connectionist models and their applications: Introduction. *Cognitive Science*, Vol. 9, No. 1.

Findler, N.V. (ed.) 1979. *Associative Networks*. New York.

Firth, J.R. 1948. Sounds and prosodies. *Transactions of the Philological Society 1948*. Oxford. Reprinted in Makkai 1972.

Gleason, H.A. Jr. 1964. The organization of language: a stratificational view. *Report on the 15th Annual (First International) Round Table Meeting on Linguistics and Language Studies*. Georgetown University Monograph Series on Languages and Linguistics, No. 17. Washington, D.C.

Gleason, H.A. Jr. 1980. Comments on William J. Sullivan's Syntax and Linguistic Semantics in Stratificational Theory. Kac, M. (ed.): *Current Syntactic Theories: Discussion Papers from the 1979 Milwaukee Syntax Conference*. Indiana University Linguistics Club. Bloomington, Indiana.

Goldsmith, J. 1976. *Autosegmental Phonology*. Indiana University Linguistics Club. Bloomington, Indiana.

Harris, Z.S. 1944. Simultaneous components in phonology. *Language* 20. Reprinted in Makkai 1972.

Hein, U. 1981. *A Short Description of CMI - a Programming System Based on Constraints*. AILAB Working Paper No. 2. Artificial Intelligence Laboratory, Software Systems Research Center, Linköping University.

Hein, U. 1982. *Constraints and Event Sequences*. Research Report LiTH-MAT-R-82-02. Software Systems Research Center, Linköping University.

Hillis, D. 1984. The connection machine: a computer architecture based on cellular automata. *Physica* 10D. Amsterdam.

Hjelmslev, L. 1961. *Prolegomena to a Theory of Language*. Madison, Wisconsin.

Hockett, C.F. 1954. Two models of grammatical description. *Word*, Vol. 10, No. 2-3.

Karttunen, L. 1983. KIMMO: A general morphological processor. *Texas Linguistic Forum*, No. 22.

Kay, M. 1985. Parsing in functional unification grammar. Dowty, D.R., L. Karttunen & A.M. Zwicky (eds.): *Natural Language Parsing*. Cambridge.

Kay, M. 1987. Nonconcatenative finite-state morphology. *Third Conference of the European Chapter of the Association for Computational Linguistics*. Copenhagen.

King, M. 1983. Transformational parsing. King, M. (ed.): *Parsing Natural Language*. London.

Koskenniemi, K. 1983. *Two-level Morphology: A General Computational Model for Word-Form Recognition and Production*. Publications of the Department of General Linguistics, University of Helsinki, No. 11.

Lamb, S. 1966. *Outline of Stratificational Grammar*. Washington, D.C.

Lockwood, D.G. 1972. *Introduction to Stratificational Linguistics*. New York.

Makkai, V.B. (ed.) 1972. *Phonological Theory*. New York.

Maleki, J. 1987. *ICONStraint, A Dependency Directed Constraint Maintenance System*. M.Sc. thesis at Linköping University. LiU-Tek-Lic 1986:11. Linköping.

McCarthy, J. 1981. A prosodic theory of nonconcatenative morphology. *Linguistic Inquiry*, Vol. 12, No. 3.

McCarthy, J. 1982. Prosodic templates, morphemic templates and morphemic tiers. van der Hulst, H. and N. Smith (eds.), *The Structure of Phonological Representations* (Part I). Dordrecht - Cinnaminson.

Mel'čuk, I.A. 1974. *Opyt teorii lingvističeskix modelej "Smysl ≪ = ≫ Tekst"*. Moskva.

Mel'čuk, I.A. 1984. *Dictionnaire explicatif et combinatoire du français contemporain (Recherches lexico-sémantiques)*. Montreal.

Mel'čuk, I.A. & N. Pertsov 1987. *Surface Syntax of English*. Amsterdam - Philadelphia.

Mel'čuk, I.A. & A.K. Žolkovskij 1984. *Tolkovo-kombinatornyj slovar' sovremennogo russkogo jazyka* /Explanatory-Combinatorial Dictionary of Contemporary Russian/. Wien.

Quillian, M.R. 1968. Semantic memory. Minsky, M. (ed.): *Semantic Information Processing*. Cambridge, Massachusetts.

Reich, P. 1969. The finiteness of natural language. *Language*, Vol. 45, No. 4.

Reich, P. 1970. *A Relational Network Model of Language Behavior*. University of Michigan Ph.D. Distributed by University Microfilms International, Ann Arbor, Michigan.

Robins, R.H. 1957. Aspects of prosodic analysis. *Proceedings of the University of Durham Philosophical Society* 1. Reprinted in Makkai 1972.

Sågvall Hein, A. to appear. Parsing by means of Uppsala Chart Processor (UCP). To appear in: Bolc, L. (ed.), *Natural Language Parsing Systems*. Berlin - New York.

de Saussure, F. 1959. *Course in General Linguistics*. Ed. by Charles Bally and Albert Sechehaye, in collaboration with Albert Riedlinger. Translated, with an introduction and notes by Wade Baskin. New York - Toronto - London.

Schnelle, H. 1981. Elements of theoretical net-linguistics. Part 1: Syntactical and morphological nets - neuro-linguistic interpretations. *Theoretical Linguistics* Vol. 8, No. 1-3.

Schreyer, R. 1980. The definition of nodes in a stratificational grammar. McCormack, W.C. & H.J. Izzo (eds.), *The Sixth LACUS Forum 1979*. Columbia, South Carolina.

Schreyer, R. 1981. The linearization of speech. Some basic problems. Copeland, J.E. & P.W. Davis (eds.), *The Seventh LACUS Forum 1980*. Columbia, South Carolina.

Sgall, P., L. Nebeský, A. Goralčíková & E. Hajičová 1969. *A Functional Approach to Syntax in Generative Description of Language*. New York.

Sowa, J.F. 1984. *Conceptual Structures*. Reading, Massachusetts.

Withgott, M. & P.-K. Halvorsen 1984. *Morphological Constraints on Scandinavian Tone Accent*. Report No. CSLI-84-11. Center for the Study of Language and Information, Stanford University.

Thrane, T. 1988. Symbolic Representation and Natural Language.

ABSTRACT

The notion of symbolizability is taken as the second requisite of computation (the first being 'algorithmizability'), and it is shown that symbols, *qua* symbols, are not symbolizable. This has farreaching consequences for the computational study of language and for AI-research in language understanding. The representation hypothesis is formulated, and its various assumptions and goals are examined. A research strategy for the computational study of natural language understanding is outlined.

SYMBOLIC REPRESENTATION AND NATURAL LANGUAGE

by

Torben Thrane

Center for Informatics, University of Copenhagen

1. Introduction: algorithms and symbolic representation

The algorithm is without doubt *the* fundamental concept in computer science. Problems that can be solved by computer are all algorithmic: they can be presented on a form which invites a division of the overall problem into constituent parts, each of which can be solved sequentially and deterministically. When the last constituent problem is solved the overall problem is solved.

To get a computer to solve a problem, however, its constituent parts must be *symbolizable:* they must be put on a form which is accessible to the computer. This precondition is summed up under the rubric 'representation'.

The inescabability of these two preconditions is never in doubt. However, doubts have recently been raised with respect to the value of the comparison that is often made, explicitly or implicitly, between the problem solving capacities of men and machines, and in particular with respect to the role allegedly played by representation as the constitutive feature of those capacities.

Understanding natural language is among the problems whose solution has been expected to be accessible through computer simulation, precisely on the assumption of a common representational basis for the problem solving capacities of men and machines.

2. The representation hypothesis

The topic of the present paper, thus loosely outlined, is *symbolic representation*, in general as well as more specifically with respect to the role it plays in computational linguistics. Initially, the notion of representation can be presented as a simple formula:

(1) **a R b**, which reads: '**a** represents **b**'.

Representation is the name of a relation holding between two entities. The logical properties of the relation are usually taken to be *irreflexivity, intransitivity,* and *asymmetry*. Apart from these logical ones, **R** has properties sometimes summed up by saying that **a** *stands for, complies with, refers to, symbolises, denotes, depicts, designates, corresponds with* **b**.

The logical properties of the entities between which representation holds are more difficult to characterize briefly. Let us assume, initially, that **a** is a physical entity, whereas **b** is typologically unspecified. We return to this issue below.

In relation to (1) we can formulate an overall hypothesis, the *representation hypothesis*, which says:

(2) All intelligent behaviour presupposes the formula (1).

Ultimately, this hypothesis aims to explain how such systems as Miller (1984) called 'informavores', can function as autonomous entities in larger physical environments, which they both affect and are affected by.

2.1. The adequacy of the formula

The formulation (1) invites the view that representation is a contextfree phenomenon, and that it eludes situationally conditioned interpretation. This is incorrect. Already C.S. Peirce - who in this connection can be considered one of the founding fathers of representation theory - insisted on the decisive influence that situation and context has on the interpretation of a sign. And perhaps even more importantly, he insisted that even the recognition of a physical entity *as* a sign presupposed background, interpretation, and what he called 'semiosis' or - as we shall say - 'the semiotic process': the process by which there is created in an observer a mental correlate - Peirce's 'interpretant' - of a physical phenomenon which, in virtue of this process, now becomes a sign *of* its object, **b**, to the observer (*CP* 2.227-9; Hookway, 1985:118-144). From this perspective, nothing is a sign 'in itself'. A sign is *created* - through the semiotic process. So the formula (1) can be amended to:

(3) **a R b** for observer **O** in situation **S** in virtue of the conventions **C**.

This means that an internal representation of **b** has been created in O, 'corresponding to' the physical sign, **a**. This internal representation, according to Peirce, is itself a sign, for which a new interpretant can be created by a recursive application of the semiotic process, and so on, ad infinitum. By way of continuation of the discussion of the logical properties of the entities between which **R** holds, we get a glimpse here of a systematic vacillation in the conception of **a** in the formula: **a** can either be regarded as the *physical sign* which represents **b**; or else **a** can be understood as the *conceptual structure* which has been created in O by virtue of O's taking **a** as a sign for **b**.

If **a** in this way can be thought of as either a physical or a mental phenomenon, **b** must be so conceived as well. It causes no trouble to entertain the idea that physical entities can be represented. Nor does it cause trouble to entertain the idea that mental or abstract phenomena can be represented. There would seem, therefore, to be no trouble in accepting that meaning can be represented, no matter whether meaning is considered to be a mental phenomenon or not; cf. Searle 1983:Ch.8 for a general discussion of this point.

However, there does crop up a problem for computational linguistics. On the view set out above, natural language is itself a representational system of interpreted symbols. At the same time, natural language is - for computational linguistists - a phenomenon that must itself be representable by computationally inter-

pretable symbols. Here the intransitivity of the general representation relation becomes apparent. If it were transitive it would be child's play to construe natural language interfaces, since then, say, a string variable, **a**, would appear to represent whatever the content of **a** represents. But this is not how things work. If **a** in this instance represents anything apart from the sequence of alphanumerical characters that make up the string, then it is a location in the computer's memory, and not, for example, the person that a string 'Tom Jones' is supposed to represent in a given context. From one perspective, then, computational representation of natural language is a special case of hypostasis.

Emerging from this discussion is the following startling fact: symbolic representation of an object, **b**, which *itself* has been interpreted as a symbol, is impossible! It can be a physical entity which - in other circumstances - *could* function as a symbol. Or, it can be a mental phenomenon, an abstract object, a conceptual structure, in addition to whatever meaning is supposed to be.

This conclusion can be summed up in slogan fashion:

(4) *Symbols are not symbolizable*

This slogan may have consequences for the proper study of various linguistic phenomena, anaphora for example. Of more immediate concern, however, is the fact that it can be construed as the basis of the bipartition which characterizes previous attempts to justify the representation hypothesis.

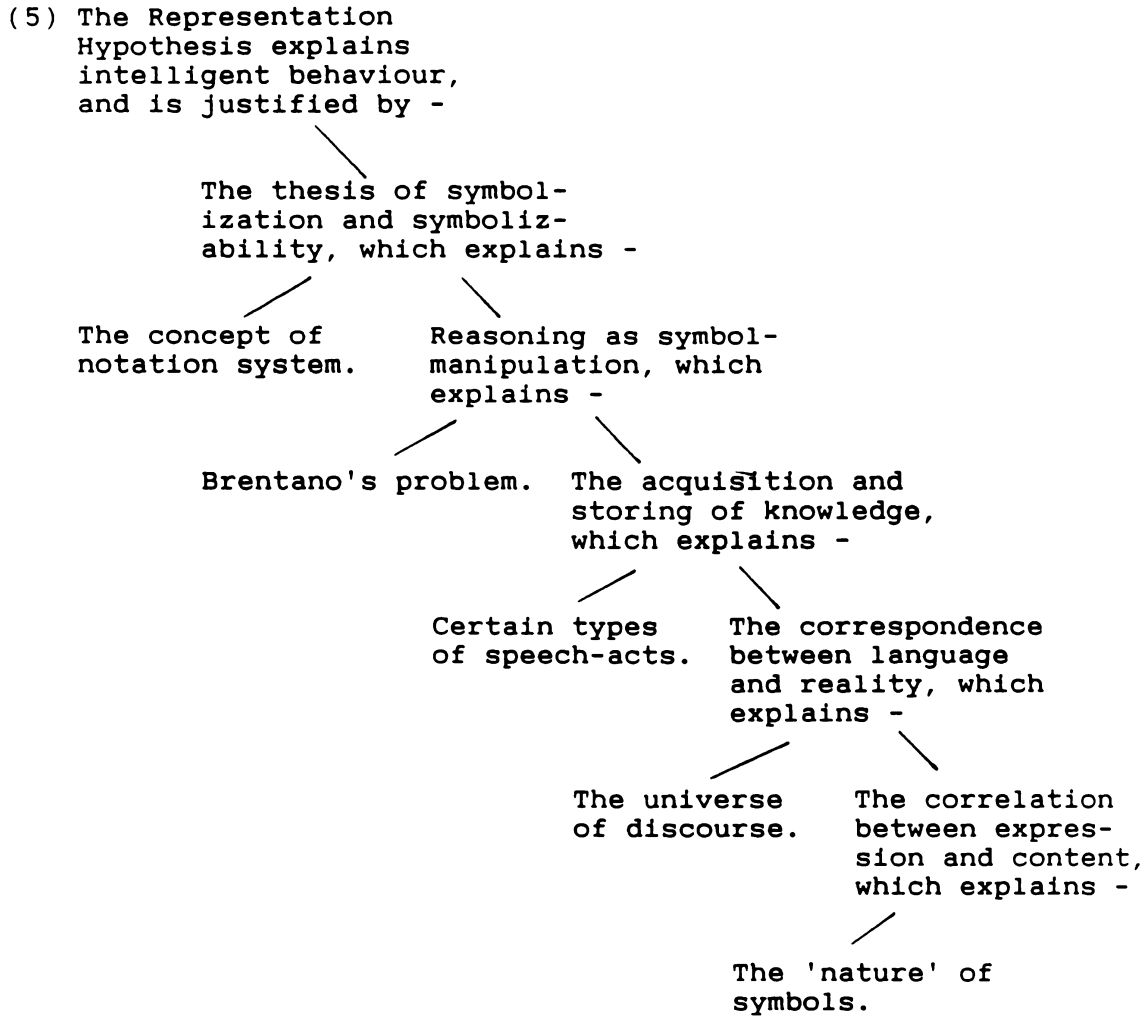
2.2. Stages along the path of the representation hypothesis

Attempts to justify the representation hypothesis come from many sides and from many more or less related academic disciplines. The same can be said of the attempts to dismiss it as untenable, in particular and most recently by Winograd & Flores (1986).

Let us first dismiss one obvious possibility of discrediting the representation hypothesis. It criticizes any attempt to justify it that takes the form of clarifying (1), correctly claiming that justification of it should take the form of a clarification of (3). Clarification of (1) enters into the ultimate clarification of (3), but they are not the same; nor can clarification of (1) ever on its own count as justification of (2). I shall therefore assume, despite some evidence to the contrary, that everybody who has been seeking justification of (2) in the pursuit of clarification of (1) in fact consciously, if tacitly, have been pursuing clarification of (3).

Clarification of (3) could be regarded as an algorithmic problem for the ultimate solution of which a number of constituent problems have been formulated and described by various academic disciplines and scholarly persuasions.

These constituent problems can with some idealization be presented in a hierarchical structure of mutually explanatory and 'support' systems, as shown in (5):


(5) The Representation Hypothesis explains intelligent behaviour, and is justified by -
The thesis of symbolization and symbolizability, which explains -
The concept of notation system.
Reasoning as symbol-manipulation, which explains -
Brentano's problem.
The acquisition and storing of knowledge, which explains -
Certain types of speech-acts.
The correspondence between language and reality, which explains -
The universe of discourse.
The correlation between expression and content, which explains -
The 'nature' of symbols.

The right-hand side comprises theories, hypotheses and presuppositions which form essential components of the overall Representation hypothesis, whereas the left-hand side displays (examples of) concrete problems or questions, the answers to which presuppose the validity of the thesis in question. In what follows I will discuss the items on the right-hand side. The items on the left-hand side will be considered in the form of a series of digressions, which together will outline a research strategy for man-machine interaction in natural language.

3. The constituent parts of the representation hypothesis

The list of constituent problems under (5) reads like a catalogue of a significant portion of Western philosophy, cognitive psychology and epistemology in terms of substance. We shall be concerned with them only from the computational point of view.

3.1. Physical symbol systems

The bipartition mentioned above, and the vacillation in the conception of **a** mentioned in 2.1., has led to a bifurcation of computational research which both proceed from Newell's (1980; Newell & Simon 1976) account of a physical symbol system.

A physical symbol system is a system which subscribes to the laws of physics and which at any given time contains a set of struc-

tured objects called 'expressions' or 'symbol structures'. Each expression is constituted by a number of *symbols*, ordered in a principled way. Symbols are physical objects whose internal structure may be quite complex, but which have the common property of being combinable with other symbols to form expressions. In addition, a physical symbol system comprises a set of processes that may create, change, destroy, and reproduce expressions. A physical symbol system, then, is a machine which produces a continuous, but continually changing, stream of symbol structures.

The precondition for the proper functioning of a physical symbol system is the notion of *interpretation*, as defined in computer science (Newell 1980:158). Interpretation is there understood as the act of accepting as input an expression which represents a process, and then executing that process.

The first of the two directions of computational research alluded to above studies the internal structure of symbols with a view to establishing a typology of symbols of greater perspicuity and more practical versatility than Peirce's, for example in connection with the generation of fonts and character sets (cf. Knuth 1982; Hofstadter 1982), the creation of MM interfaces, document representation (cf. Levy, Brotsky & Olson 1987; Southall 1987), the development of graphics systems, etc. This direction, under the label 'figural representation', is deeply influenced by the art-aesthetic discussion of the concept of representation, which rejects *similarity* between **a** and **b** as the proper grounds for representation in favour of substitution or functional equivalence. Consider in this connection Picasso's famous reply to contemporary criticism that his portrait, *Gertrude*, did not resemble Mrs. Stein: "Don't worry. It will".

The other direction seeks to explain the behaviour of informavores. This direction proceeds from cognitive psychology as much as from semiotics - it is known as 'cognitive science' which subsumes the more practical study of artificial intelligence. This was Newell's own main interest. He formulates the following hypothesis:

(6) *The Physical Symbol System Hypothesis*

> The necessary and sufficient conditions for a physical system to exhibit general intelligent action is that it be a physical symbol system.
>
> Newell, 1980:170

'Necessary' and 'sufficient' in this connection mean, respectively, that a system displaying what we would be prepared to call intelligent behaviour, will always upon closer inspection turn out to be a physical symbol system; and a physical symbol system of sufficient size will always be amenable to organization in such a way that it will display behaviour that we would be prepared to call intelligent.

Clearly, interest in the properties of the symbol differs according to which of the two directions one follows. In the first instance we get an interest in the physical properties of symbols that may alleviate their *extensional* interpretation. In the second, interest centres around the physical properties of symbols that will secure a particular *intensional* interpretation. In these terms the semiotic process can be seen as a complex series

of steps whereby a particular physical object undergoes various internal processes (Newell calls them 'symbolic'), whereby they are turned into meaningful structures, *ie* structures that determine the symbol system's subsequent behaviour. It is important to realize that we have no immediate access to these structures, but only recognize them through the behaviour of the system. Consequently, we can only try to describe them on the basis of an abstraction from observed behaviour, in an appropriate formal notation, if the need arises. In this way, both directions take a crucial interest in the physical properties of symbols which converges in the need for interpretation, extensional and intensional. This leads to a digression on notational systems.

3.1.1. Digression: Notational systems

The fundamental property of a symbol is that it is an object manifest to the senses. But - as Newell made clear - a symbol may be of a complex internal structure. This structure will in some cases be amenable to description by means of a notational system, *viz* those cases where the atomic parts of every symbol in the scheme constitute a set that satisfies the five requirements on notational systems formulated by Nelson Goodman (1976):

(7)(a) *Syntactic discreteness:* the decision whether some arbitrary inscription belongs to a particular character and not another is deterministic;

(b) *Syntactic disjointness:* any inscription either belongs to one and only one character, or does not belong to the scheme;

(c) *Unambiguity:* any character, as well as any inscription of any character, must be unambiguous;

(d) *Semantic differentiation:* the decision whether some referent of a particular inscription of any character belongs to one class of objects or another is deterministic;

(e) *Semantic disjointness:* no two characters in a notational system can have any referent in common.

In (8) are displayed samples of signs that are amenable to internal description on the basis of a notational system (a), and of signs the internal structure of which does not reflect a notational system, but which must rather be described on the basis of iconicity (b):

(8)(a)

/* Insert page with figures (8)(a) and (b) and remove this line*/

(b)

The interesting thing about these claims in our connection is that the alphabet satisfies them, whereas larger linguistic entities as a rule do not. The *International Phonetics Association* notation in fact subsumes both types: a subset - used for phonemic transcriptions - forms a notational system on the criteria above, the scheme as a whole - used for phonetic description - does not. Semantic networks, frames, etc., do not constitute notational systems in the required sense, the propositional and predicate calculi do. And finally, all (procedural?) programming languages are notational systems. The parenthesis indicates some hesitation with respect to programming languages like Prolog and Lisp, and many tools specifically developed as system-building aids for knowledge engineering. And the hesitation is due to uncertainty whether to regard such languages from the point of view of the programmer or from the point of view of the machine in making the decision. This ties in with the representation hierarchy (see below, 3.2.1).

3.2. Reasoning as symbolmanipulation

> ..Reason, *when wee reckon it amongst the Faculties of the mind ... is nothing but* Reckoning *(that is Adding and Subtracting) of the Consequences of generall names agreed upon, for the* marking *and* signifying *of our thoughts.*

This passage, from Thomas Hobbes *Leviathan* (1651:I.5), is one of the earliest expressions of what Winograd & Flores somewhat sweepingly style the 'rationalistic tradition' in representation theory. In our own time, Johnson-Laird (1983:2-4) credits Kenneth Craik with the first full-fledged modern version. He describes reasoning as a process that falls into three phases:

(9)(a) A 'translation' of some external process into an internal representation in terms of words, numbers or other symbols;

(b) The derivation of other symbols from them by some sort of inferential process;

(c) A 'retranslation' of these symbols into actions, or at least a recognition of the correspondence between these symbols and external events, as in realizing that a prediction is fulfilled.

Johnson-Laird (1983:2-3)

The symbol has become a mental code with psychological reality, a view which harks back to Peirce's notion of 'interpretant'. However, we should be wary of identifying the two views, mainly because cognitive scientist are more interested than Peirce in explaining the *behaviour* of a system on the basis of representational (semantic) content; cf. Pylyshyn's (1984:39) formulation:

> In cognitive science,..., we want something stronger than derived semantics, *inasmuch as we want to explain the system's behavior by appealing to the content of its representations.*
>
> (my italics)

The justification of this thesis is central to cognitive psychology, for if it can be justified, Brentano's problem disappears.

3.2.1. Digression: Brentano's problem

How can physical stimuli determine the behaviour of a biological system *even though* no direct causal relationship exists between stimulus and behaviour? This is a nutshell formulation of the classical problem which Brentano attempted to solve by introducing a distinction between an 'object' and our mental 'representation' of it, and which, since then, has been one of the constitutive problems of cognitive psychology, on a par with philosophy's mind-body problem.

Rather than solve it, cognitive science believes to have *dissolved* it - with reference to the physical embodiment of symbol systems and the representation hierarchy (Newell 1980:172-75, 1982; Haugeland 1982; Johnson-Laird 1983:399ff; Pylyshyn 1984; but cf. also Winograd & Flores 1986:86-89 and Ch. 8).

A computer can be exhaustively described in various ways: as a physical device, as a collection of logically interpreted circuits, as a deterministic sequence of states defined by a program, etc. The notion of a hierarchy of representational levels stems from the realization that, depending on which type of description is chosen, there is a corresponding, radical change in the proper description of the nature of the information-processing relevant to that type: as electrical current switching on

and off, as interpretation of particular electrical patterns as logical values, as interpretation of larger chunks of such patterns as alphabetic characters or numbers, of yet larger chunks as lists of objects or strings of characters, etc.

There is no disagreement on these principles in so far as they are used to describe what *computers* do. But if the account is used as a metaphor - or, indeed, as a literal description - of what *people* do in processing information from physical stimuli of the senses, disagreement is fierce. Searle (1980) draws a useful distinction between 'weak' and 'strong' artificial intelligence research in a discussion of these matters. Weak AI is characterized by regarding computers and programs as tools that enable us to test hypotheses about cognitive processes in a rigorous manner, whereas - in strong AI - the appropriately programmed computer provides the explanation of such processes, with a significant parallel being posited between the jump from hardware to software (in the case of the computer) and from brain to mind (in the case of people). But - to the strong AI researcher (*eg* Pylyshyn) - the point is precisely that it is impossible to define a definite line across which this jump is made, as suggested by the fluid, yet perfectly specifiable, levels of the representation hierarchy.

The representation hypothesis, in conjunction with the strong AI interpretation of the representation hierarchy, provides a sufficiently rich explanation of our interaction with our environment to merit serious consideration. Attempts to discredit it, therefore, must provide a plausible, and equally rich, alternative solution to Brentano's problem to be creditable themselves. This

is what Winograd & Flores (1986:Ch. 4) do in their appeal to the cognitive theories of the Chilean biologist Maturana.

3.3. The acquisition and storing of knowledge

The problem of how we *acquire* knowledge is at the epistemological core of Western philosophy. The problem of how we *store* it, once acquired, has only become of prime importance with the advent of computers, for if there is one thing on which all AI researchers agree it is that knowledge is extremely bulky. So, although the question of how best to feed the necessary information into the system is of concern to AI research in general, the burning questions are rather how to cope with its bulk without loss, and how to preserve it in a form suitable for rapid access and retrieval. These problems crucially involve matters of representation, and specific *knowledge representation languages* have been developed to cope with them; cf. Waterman (1986:339-365) for a survey.

All attempts to create knowledge representation languages have assumed the validity of the *knowledge representation hypothsis*, first explicitly formulated by Smith (1982). It goes like this:

(10) Any mechanically embodied intelligent process will be comprised of *structural ingredients* that a) we as *external* observers *naturally* take *to represent a propositional account of the knowledge that the overall process exhibits*, and b) independent of such external semantic attribution, play a *formal but causal* and essential role in engendering *the behaviour that manifests that knowledge*.

Qu.f. Brachman & Levesque (1985:33)

(my italics)

This formulation comprises a series of quite central claims about the nature, organization, and function of the knowledge required for the performance of rational behaviour.

Firstly, knowledge is *representable* - or symbolizable - by *structural* elements. Accessible knowledge is assumed to be organized along previously determined patterns or principles. Only accessible knowledge determines behaviour.

Secondly, the representation of knowledge structured along these lines is assumed to consist of a collection of *propositions*, which we can define here as truth-valued abstractions over states of affairs.

Thirdly, it is supposed to be *natural* for us to see the situation in this light.

Fourthly - and in direct continuation of the digression on Brentano's problem above - it is the knowledge, structured in this way, that is the *cause* of the system's behaviour, and which we call intelligent because it reflects a reasonable 'awareness' of the accessible knowledge. Behaviour is the only external (or interpersonal) evidence of accessible knowledge.

Finally, the triggering of rational behaviour is strictly formal, which means that it is not the propositional *content* as such which is the factor determining behaviour, but rather the structural occurrence of a particular configuration of truth-values in the overall knowledge structure. Pylyshyn's (1984) major aim is to escape this conclusion.

No doubt all of these claims - and their consequences - merit discussion, but I will stick to just one, viz. that acquired knowledge is structured propositionally, and I will do so by way of yet another digression, on speech acts.

3.3.1. Digression: Speech Acts

To make a statement, to ask a question, to issue an order are the three major types of speech act, in the sense that most languages make distinctions in their grammatical systems between the types of sentences typically used to perform them: declarative, interrogative, and imperative, respectively. Among these, declarative sentences have had a particularly prominent position in theoretical discussions of semantics, because they are natural language expressions of streamlined, truth-valued propositions, and because theoretical semantics has often seen it as its major business to account for the conditions that make a particular declarative sentence true.

If, however, the major semantic business is to account for the circumstances in which a particular sentence can be said to have been *understood*, priorities change. Documentation of understanding is *action:* documentation of the understanding of a question is a suitable locutionary act, documentation of an order is execution of a suitable physical act, locutionary or not. These are fairly clear. But what action do we perform in order to document understanding of a declarative sentence?

One possible answer supports the claim above, that knowledge is stored and structured on propositional form. Informally, it says that anyone who hears a declarative sentence will carry through a recursive check as to whether the proposition expressed by it is already part of his 'knowledge base' or not. If it is, and if there is no new evidence for altering one's knowledge on this point, the sentence is dismissed. If it is not, the propositional content of the sentence is checked for *consistency* relative to the knowledge base. If it is consistent, the knowledge base is updated with the new information. If it is not, an assessment is made as to whether a revision of existing knowledge is called for in the light of the new information, or whether the new information is 'wrong', given the validity of the knowledge base. In the former case, the entire knowledge base is revised, including the set of possible inferences that can be drawn from it. In the latter case, the new information is again rejected (or disputed). This, in barest outline, is the mechanism that Johnson-Laird (1983:Ch.15) takes as the foundation of his theory of how we create 'mental models' of our environment.

3.4. The relationship between language and reality

If the representation hypothesis as formulated in (2) is convincing, then its inherent thesis of the relationship between language and world must be too. Winograd & Flores describes the latter thus:

(11) The rationalistic tradition regards language as a system of symbols that are composed into patterns that stand for things in the world. Sentences can represent the world truly or falsely, coherently or incoherently, but their ultimate grounding is in their *correspondence* with the states of affairs they represent. This concept of correspondence can be summarized as:

1. Sentences say things about the world, and can be either true or false;
2. What a sentence says about the world is a function of the words it contains and the structures into which these are combined;
3. The content words of a sentence (such as its nouns, verbs, and adjectives) can be taken as denoting (in the world) objects, properties, relationships, or sets of these.

This is a somewhat simplified account, which I will attempt to show by means of a slightly expanded paraphrase. Sentences are said to *represent* 'states of affairs', because they say something about how the world is organized. If the state of affairs that a sentence represents can be found in the world, then the sentence is true, otherwise false. A state of affairs is a delimited collection of objects which have particular properties and between which particular relations hold. A sentence represents a state of affairs just in case the words or phrases of the sentence, and the structure of which they are a part, refer to those objects

that make up the state of affairs, and specify their properties and mutual relations.

No wonder that Winograd & Flores balk at this. The paraphrase implies that there is permanence to a 'true' way in which a sentence represents a state of affairs, that a state of affairs can be identified as the one 'missed' by a particular sentence, that there is an *a priori* global but finite set of objects, properties and relations which we all share a common knowledge of, and which we all have equal (great or small) possibilities of describing 'correctly'. O'Connor (1975) provides further evidence of the insufficiency of the above account of the correspondence theory of truth.

However, a thesis of a *correspondence* relation between language and world need not be tied to such a restrictive formulation. There is nothing in a correspondence thesis that prevents reference to the utterance situation as the constitutive element of linguistic communication. There is nothing to prevent a general account that incorporates speech act theory and correspondence theory. And there is absolutely nothing to prevent us from rejecting the simplistic idea of dependency between states of affairs and linguistic expressions that forms part of the above characterization. This leads to the next digression, on universes of discourse.

3.4.1. Digression: The universe of discourse

Proposition 5.6. in Wittgenstein's *Tractatus* states: "*Die Grenzen meiner Sprache* bedeuten die Grenzen meiner Welt". A possible in-

terpretation of this proposition is that language is what constitutes our world. There are other possible interpretations. I shall provide one more below.

The current interpretation leads to the assumption that the world which language immediately constitutes, is not the actual, physical world, but rather an abstract, a model, or - as we shall call it - a *universe of discourse* or, equivalently, a *discourse universe*.

Discourse universes are private universes, but we can share one or more of them, in part. Long married couples - who are often held to be capable of wordless communication - will, in this jargon, share a large portion of their overall universe of discourse. Universes of discourse may be large and small, and may be more or less densely populated. We can operate on the basis of more than one universe of discourse at the same time, and we can shift from one to another with perfect ease between conversations and even during conversations. And lastly, universes of discourse are intentional, in Searle's (1983:1) sense of 'intentionality':

> Intentionality is that property of many mental states and events by which they are directed at or about or of objects and states of affairs in the world.

The main idea is that language use implies continuous creation, revision, and deletion of universes of discourse for the participants of the conversation. The overall linguistic mechanisms for these purposes are what I have previously styled 'the referential properties of language' (Thrane 1980). Revision and updating of the current discourse universe is in the main associated

with identitive, generic and qualitative features (determining definiteness, specificness, certain aspects of conditionality, and referent typology), whereas shifts between universes of discourse are typically associated with presentative and partitive features (determining discourse 'scope', various aspects of definite and indefinite quantification, and other aspects of conditionality). The conditionality of referential expressions is the general mechanism for establishing the 'laws' that hold in a universe of discourse.

On this view, an utterance becomes a *symptom* of the state of the speaker's current universe of discourse, where 'symptom' is intended in the technical sense of Lyons (1977:108) as 'a sign or signal which indicates to the receiver that the sender is in a particular state'. It is incumbent on the receiver, on the basis of his interpretation of the symptom, to gain insight in the state, perhaps to adapt his own universe of discourse accordingly, or to try to persuade the speaker to revise his. Mutual understanding can, under the same view, be regarded as a progressional striving towards the greatest possible congruence between the current discourse universes of speaker and hearer, through cooperation and negotiation, for example about the proper definition or interpretation of a word, or determination of the reference of an expression.

The correspondence relation which is being championed here is a function from a universe of discours to a state of affairs. And accepted truths are those special cases in which the universes of discourse of many (and hopefully, of all) map into the same state of affairs. This does not prevent 'truth' from being the property

of just one person - in the sense that the relevant state of his universe of discourse may map into a state of affairs that, over time, will be mapped by the universe of discourse of society at large. This is the only kind of 'absolute truth' that the views taken here will sanction.

3.5. The correlation between expression and content

The last step in this account of the various stages of the representation hypothesis is the question of the connection between expression and content, or meaning. The two currently most favoured bids as to what meaning is, derive from model-theoretic semantics and Situation semantics. Both attempt to characterize meaning in a way that enables them to explain the relation between language and reality, both set off from Frege's distinction between sense (intension) and reference (extension), and both avail themselves of a logical formalism to represent meaning. But from this point on they part company.

Model-theoretic semantics has taken over Leibniz' notion of 'possible worlds', and defines truth relative to that. The intension of a sentence is a function from possible worlds to truth-values, whereas the intension of terms and predicates is a function from possible worlds to, respectively, objects and sets. The extension of a sentence is a truth-value, whereas the extension of terms and predicates is, respectively, objects and sets. So the meaning (intension) of a linguistic expression is those properties of the expression which in any conceivable situation determine what language external objects or sets the expression refers to in that situation, or - if the expression is a sentence - if the

expression is true in the situation. More briefly: the intension of a sentence is the set of conditions that has to be satisfied in some possible world for the sentence to be true in that world. Model-theretic semantics is fundamentally intensional, and it represents intensions by means of predicate calculus formulae.

In Situation semantics (Barwise & Perry 1983), meaning is a relation between types of situations. This view stems from the recognition that although there is a principled difference between 'natural' carriers of meaning ('signs'), and 'unnatural' or 'conventional' carriers of meaning ('symbols') - illustrated by the difference between 'smoke means that something is burning' and '"something is burning" means that something is burning' - then both instances involve the transmission of information. Utterance situations, in other words, belong to a type of situation in which the transmission of information is based on symbols and their interpretation. Situation semantics is fundamentally extensional, and its representation of meaning is in fact a representation of situation types, in a formal set theoretic notation.

Even though both model-theoretic and Situation semantics seek to explain the relationship between language and 'the world' through the development of formal notations for the meaning of natural language, both theories suddenly lose sight of language. The model-theoretic solution to the overall problem has the consequence that meaning is divorced from language. If intension is a function from possible worlds (or, in more recent developments, states of affairs indexed for time) to extensions, then the linguistic expression has disappeared and must be reintroduced by a

general interpretation function from expressions to intensions. The Situation semantic solution fails to draw what to me appears to be a basic typological distinction between utterance situations (situations *created* by the making of an utterance), and other situations (typically situations *described* by making an utterance). The distinction is based on the notion of intentionality. Utterance situations occur whenever utterances are made; but utterances are the outward manifestation of intentional states of various sorts. Described situations, on the other hand, are not intentional. They are rather 'extentional' in the sense of being the goals at which some intentional states are directed.

Quite apart from such theory-dependent problems, the two semantic theories share a common problem with any other theory that subsumes attempts to represent natural language meaning by means of a formal notation. The creation of any formal representation of the meaning of a natural language sentence amounts to nothing but a claim of synonymy between two material expressions, of course. The problem that must be faced by all semantic theories that rely on formal representations of meaning, is that of interpretability. Even if the formal representation meets all requirements of syntactic well-formedness, internal consistency, etc., it is, in the last resort and in principle, *only* interpretable through the natural language sentence with which it is claimed to be synonymous. The problem of interpretability stems from the non-symbolizability of symbols, *qua* symbols, commented on above (2.1). I take this to be the onus of the second interpretation of Wittgenstein's famous proposition, promised above: my world is only *accessible* through my language. The consequence of this interpretation is that natural language is in fact the *only* efficient meaning representation schema!

3.5.1. Digression: Computational meaning representation

Does this conclusion mean that the representation hypothesis has foundered as a serious explanatary basis for communication, reasoning, and knowledge organization? I don't think so. For even if two well-merited and well-developed semantic theories face problems with respect to their capacity for giving a global characterization of the meaning of natural language sentences, both have developed techniques and insights that can be brought to bear on concrete projects that aim at exploring man-machine communication in natural language. This has never been the primary motivation for model-theoretic semantics nor for Situation semantics.

Such a narrower approach to the problem will have to set off from a set of assumptions specifically geared to, and delimiting, its scope.

First of all, a distinction parallel to Searle's distinction between 'weak' and 'strong' artificial intelligence is called for within the computational study of language. 'Weak', or 'objective' computational linguistics is characterized by

- regarding language as an object of study;
- regarding the computer as a tool;
- regarding a program as a hypothesis of language structure.

'Strong', or 'subjective', computational linguistics, in contrast, is characterized by

- regarding language as a medium of communication;
- regarding the computer as a partner in communication;
- regarding a program as a hypothesis of communicative interaction.

The ultimate goal of the project, on this distinction, falls within 'strong' computational linguistics - and it is still a moot point, to what detailed extent, and to what heuristic level, 'weak' computational methods should enter into it (nature of the parsing required, level of morphological refinement, etc.).

Secondly, it is a limitation that computers, on the 'strong' view above, meaningfully enter into utterance situations only, indeed, utterance situations of an impoverished kind: there can (so far) be no reliance on suprasegmentals, no reliance on gestural or facial information, there is a limited range of language functions, etc. Thus, 'addressing' a computer is, invariably, an attempt to gain access to its universe of discourse, either with a view to changing it or to get documentation of its current state in the form of an appropriate responsive action. If this action is to be based on the computer's 'understanding' of the meaning of a natural language input, meaning in this context is a function from an utterance situation into a discourse universe. This assumption trades on a combination of the functional and relational views of meaning characterizing, respectively, model-theoretic and Situation semantics. It further enhances referential and conative meaning, but deliberately leaves out of account aspects of emotive, phatic, metalingual, and poetic meaning.

Thirdly, the process of 'understanding' is further segmented into a process of 'deciphering' and one of 'interpreting', in the following way: 'deciphering' is a function from an *utterance* to a universe of discourse on the grounds of the 'code', whereas 'interpretation' is a function from one universe of discourse to another. 'Decipherment' will yield a rough approximation to what the hearer takes as the speaker's current universe of discourse, 'interpretation' will work on this to yield a universe of discourse more finegrained, and reflecting the hearer's conception of what the speaker 'has in mind'. Matters of poorly understood words or phrases will be dealt with by 'decipherment', matters of failing reference, inconsistency, etc. by 'interpretation'. 'Interpretation' in this framework is closely akin to the computational view of it (above, 3.1.), in that it involves one or more processes to be executed by the content of a universe of discourse.

Finally, as already mentioned, extension is regarded as a function from discourse universes to *described* situations. Whether the computer in this connection can be said to possess a 'true' image of the world is a question which is in principle no different from the question whether *we* can: it depends on whether our universe of discourse maps into a factual situation. And this in turn depends on the nature, quantity and quality of the *knowledge* we had access to during its establishment.

4. Final remarks

It has not been my primary concern in this paper to try to falsify the representation hypothesis, which I personally find attractive. It has been my concern, on the other hand, to present a series of aspects, interpretations, and consequences of it which in due course may *make* it falsifiable. For if it turns out that the more restrictive semantic program outlined throughout the last three digressions can be carried through, *without* any indication that the same semantic processes characterize interpersonal communication, then there is reason to believe that the representation hypothesis as formulated in (2) is either too strict or wrong. However, one of the results may well be acceptance of the view that the 'nature' of symbols is to be found among the class of topics of which Wittgenstein said:

Wovon man nicht sprechen kann, darüber muss man schweigen.

One of the fascinations about natural language, however, is that it is extremely difficult not to use it, even for discussion of topics that can't be discussed.

REFERENCES

Barwise, J. & Perry, J. 1983. *Situations and Attitudes.* MIT: Cambr. Mass. 2nd Printing 1984.

Brachman, R.J. & Levesque, H.J. (eds.) 1985. *Readings in Knowledge Representation.* Morgan Kaufmann: Los Altos.

CP = *The Collected Papers of Charles Sanders Peirce.* Edited by Charles Hartshorne and Paul Weiss. Cambr. Mass. 1935-66. (References in text are to Volume and Paragraph)

Deuchar, M. 1984. *British Sign Language.* Routledge & Kegan Paul: London.

Goodman, N. 1975. *Languages of Art.* MIT: Cambr. Mass.

Haugeland, J. 1981. Semantic Engines: An Introduction to Mind Design. In Haugeland, J. (ed.) *Mind Design.* MIT: Cambr. Mass. 3rd Printing 1985, pp. 1-34.

Hobbes. T. 1651. *Leviathan.* Pelican Books: Harmondsworth 1963.

Hofstadter, D.R. 1982. Metafont, Metamathematics, and Metaphysics: Comments on Donald Knuth's Article "The Concept of a Meta-Font". In Hofstadter, D.R. *Metamagical Themas: Questing for the Essence of Mind and Pattern.* Bantham Books 1986, pp. 260-287.

Hookway, C. 1985. *Peirce*. Routledge & Kegan Paul: London.

Johnson-Laird, P. 1983. *Mental Models*. Harvard UP: Boston.

Knuth, D. 1982. The Concept of a Meta-Font. *Visible Language* 16, 3-27.

Levy, D.M., Brotsky, D.C. & Olson, K.R. 1987. Formalizing the Figural: Aspects of a Foundation for Document Manipulation. (Draft). Intelligent Systems Laboratory, Xerox Palo Alto Research Centre. 3333 Coyote Hill Road. Palo Alto, Ca. 94304.

Lyons, J. 1977. Semantics. Cambridge UP: Cambridge.

Newell, A. 1980. Physical Symbol Systems. *Cognitive Science* 4, 135-183.

Newell, A. 1982. The Knowledge Level. *Artificial Intelligence* 18, 87-127.

Newell, A. & Simon, H.A. 1976. Computer Science as Empirical Inquiry: Symbols and Search. In Haugeland, J. (ed.) *Mind Design*. MIT: Cambr. Mass., 3rd Printing 1985, pp. 35-66.

O'Connor, D.J. 1975. *The correspondence theory of truth*. Hutchinson: London.

Pylyshyn, Z.W. 1984. *Computation and Cognition: Toward a Foundation for Cognitive Science*. MIT: Cambr. Mass., 2nd Edition 1985.

Searle, J. 1980. Minds, Brains, and Programs. In Haugeland, J. (ed.) *Mind Design*. MIT: Cambr. Mass., 3rd Printing 1985, pp. 35-66.

Searle, J. 1983. *Intentionality*. Cambridge UP: Cambridge, 5th Printing 1987.

Southall, R. 1987. A basis for the description of machine-written documents. (Draft). Intelligent Systems Laboratory, Xerox Palo Alto Research Centre. 3333 Coyote Hill Road. Palo Alto, Ca. 94304.

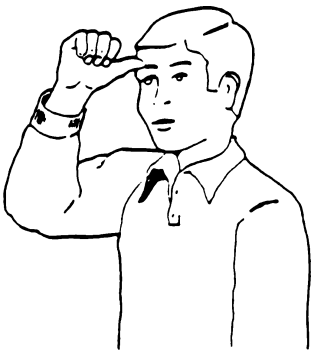
Thrane, Torben 1980. *Referential-semantic analysis: Aspects of a theory of linguistic reference*. Cambridge UP: Cambridge.

Waterman, D.T. 1986. *A Guide to Expert Systems*. Addison-Wesley: Reading, Ma.

Winograd, T. & Flores, F. 1986. *Understanding Computers and Cognition. A New Foundation for Design.* Ablex: Norwood, N.J.

Wittgenstein, Ludwig 1921. *Tractatus Logico-Philosophicus.* Routledge & Kegan Paul: London, 1966.

THINK
tab: forehead
dez: index finger extended from closed fist
sig: contact with tab



KNOW
tab: forehead
dez: thumb extended from closed fist
sig: contact with tab

CLEVER
tab: forehead
dez: thumb from closed fist
sig: movement from right to left in contact with tab

Fig. 8a *The structure of BSL signs*

FROM Deuchar (1984)

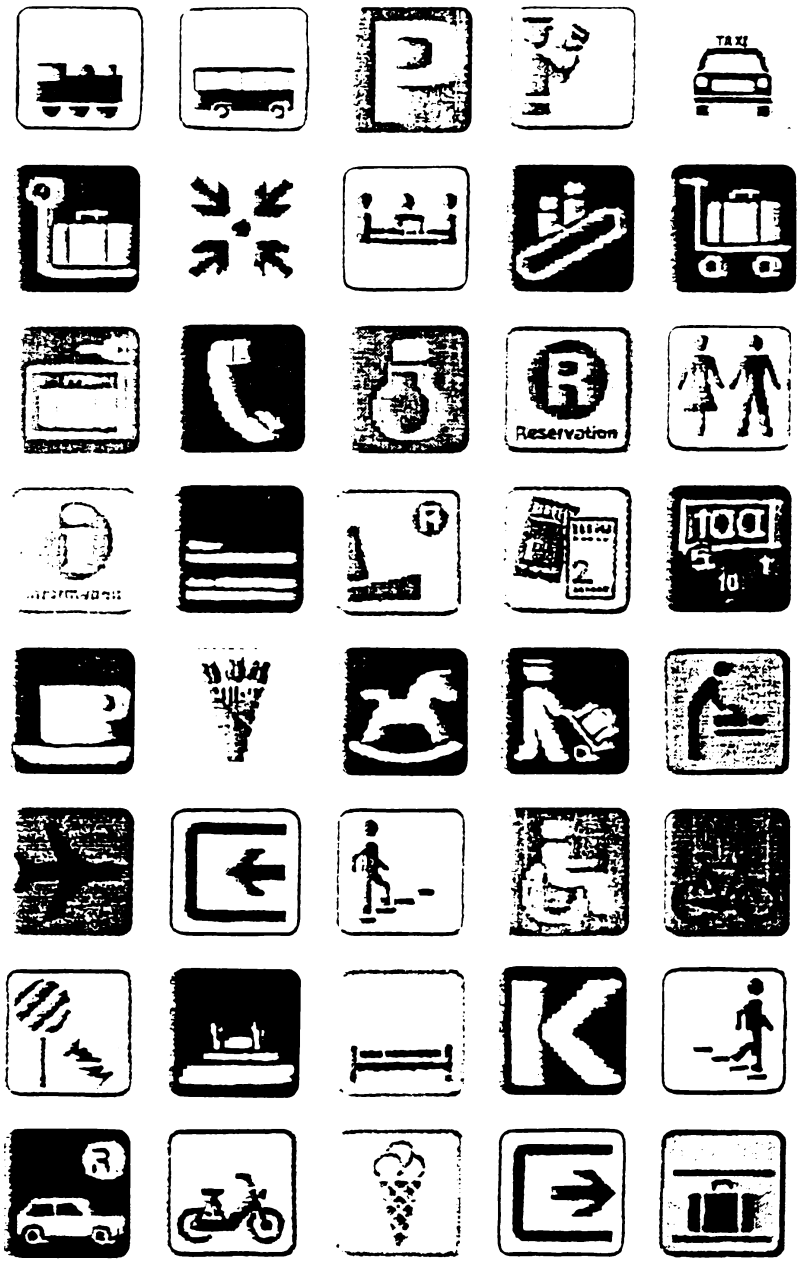


Fig. 86 Icons
FROM DSB Corporate Identity 1990.

PROCESSING SENTENCES CLAUSE BY CLAUSE

Eva I. Ejerhed
University of Umeå
Department of Linguistics
University of Umeå
S-90187 Umeå, Sweden

ABSTRACT

The paper presents and compares two different methods of parsing, a regular expression method and a stochastic method, with respect to their success in identifying basic clauses in unrestricted English text. These methods of parsing were developed in order to be applied to the task of improving the detection of large prosodic units in the Bell Labs text-to-speech system, and were so applied experimentally. The paper also discusses the notion of basic clause that was defined as the parsing target. The result of a comparison of the error rates of the two parsing methods in the recognition of basic clauses showed that there was a 13% error rate for the regular expression method and a 6.5% error rate for the stochastic method.

1. Introduction.

The present paper describes the procedure that was followed in an extended experiment to reliably find basic surface clauses in unrestricted English text, using various combinations of finitary and stochastic methods. The purpose was to make some improvements in the detection and treatment of large prosodic units above the level of fgroups in the Bell Labs text-to-speech system. This system currently relies exclusively on punctuation (commas and periods) for the detection of such units, i.e. tonal minor and major phrases. Commas are correlated with tonal minor phrases, and sentence final periods with tonal major phrases. The notion of fgroup (one or more function words followed by one or more content words), and its implementation in the Bell Labs text-to-speech system is described in Liberman & Buchsbaum (1985).

Correct automatic detection of major syntactic boundaries, in particular clause boundaries, is a prerequisite for automatic insertion of final lengthening, boundary tones and pauses at such boundaries within sentences (cf. Allen, Hunnicutt & Klatt 1987,and Altenberg 1987). These prosodic phenomena make significant contribution to the naturalness and intelligibility of synthetic speech. Unfortunately, the task of parsing unrestricted text correctly, in order to find the relevant sentence internal syntactic boundaries has turned out to be very difficult. This paper is a report of an attempt to provide a better foundation for parsing text by the use of simple finitary and stochastic computational methods. These simple methods have not figured prominently in the theory and practice of natural langauge parsing, with some exceptions (Langendoen 1975, Church 1982, Ejerhed & Church 1983). For an experimental, and more complicated method to derive all

prosodic units in the text-to-speech system, i.e. not just tonal minor and major phrases but every type of prosodic unit, from the syntactic structure and length of constituents, see Wright, Bachenko & Fitzpatrick (1986).

The first purpose of the experiment was to test the performance of a finite state parser, when the parser was given the rather difficult and substantive tasks of finding basic, non-recursive clauses in continuous text, in which each word had been tagged with a part of speech label. Parts of the tagged Brown corpus were used, representing the genres of both informative and imaginative prose. The clause grammar, consisting of a regular expression for clauses of different kinds, was constructed by the author and it was first applied to text that was guaranteed to have correct parts of speech assigned to the words, so that problems in constructing the grammar could be isolated from problems in assigning correct parts of speech. The finite state parser that used the clause grammar consisted of a program that matched regular expressions for clauses against the longest substrings of tagged words that fit them, and it was constructed and implemented by K. Church.

The second purpose was to see whether basic clauses could also be recognized by stochastic programs, after these had been trained on suitable training material. The training material was prepared by hand-correcting the output of the program that processed the regular expressions for clauses. A stochastic program for assigning unique part of speech tags to words in unrestricted text had been created by K. Church, and trained on the tagged Brown corpus (see Church 1987). The resultant program is 95-99% correct in its performance, depending on the criteria of correctness used, and it can be used as a lexical front end to any kind of parser, i.e. not necessarily stochastic or finite state parsers. However, the question presented itself whether the stochastic procedure that was so successful in recognizing parts of speech could also be applied to more advanced tasks such as recognizing noun phrases and clauses. The present paper concentrates on the parsing of basic clauses. The parsing of noun phrases by the same two methods is compared in Ejerhed (1987), and the stochastic parsing of noun phrases is described in detail in Church (1988).

The structure of the paper is as follows. Section 2 defines the target of a basic clause, and reports on the outcome of the search for such units by the two methods. Section 3 discusses the correlations between clause units as defined by this paper, and the prosodic units of tonal minor and major phrases in the Bell Labs text-to-speech system.

2. Finding Clauses.

2.1 Why Clauses?

Syntactic surface clauses are interesting units of language processing for a variety of reasons. In the surface clause, criteria of form and meaning converge to guarantee both that it can be recognized solely by surface syntactic properties

and that it constitutes a meaningful unit (ideally a proposition) in a semantic representation.

Clauses have been investigated in psycholinguistic research. Jarvella (1971) found effects of both sentence boundaries and clause boundaries in recall of spoken complex sentences and took them, along with previous results of Jarvella & Pisoni (1970), to support a clause-by-clause view of within-sentence processing.

Later research on reading comprehension has found effects on gaze duration not only of word length and word frequency, but also of syntactic local ambiguity (garden paths) and of ends of sentences (Just & Carpenter 1984). However, the study of clause units as distinct from sentence units has not been carried out systematically in psycholinguistic experiments so far, and a lot of basic facts remain to be found out about the role of clause units of different kinds in the processes whereby spoken and written language is comprehended.

2.2 The Definition of A Basic Clause.

Finding basic noun phrases is important as a stepping stone to finding clauses, on the assumption that an important subset of them have an initial sequence consisting of a noun phrase followed by a tensed verb as a defining characteristic. The result of scoring the respective success of the two methods of parsing basic noun phrases in sample text portions, reported in Ejerhed (1987), was the following. The regular expression output had 6 errors in 185 noun phrases, i.e. a 3.3% error rate. The stochastic output had 3 errors in 218 noun phrases, i.e. a 1.4% error rate. Both results must be considered good in the absolute sense of an automatic analysis of unrestricted text, but the stochastic method has a clear advantage over the regular expression method. Basic noun phrases can be found, which is of important for clause recognition.

The definition of basic clause that was used in this study has the following characteristics: a) it concentrates on certain defining characteristics present at the beginnings of clauses; b) it follows from a particular hypothesis about syntactic working memory: that it is limited to processing one clause at the time; and c) it assumes that the recognition of any beginning of a clause automatically leads to the syntactic closure of the previous clause.

It should be clear from the above, that the theoretical reasons for pursuing a recursion-free definition of a basic clause have to do with a theory of linguistic performance, rather than with a theory of linguistic competence, in which memory limitations play no part. It is a hypothesis of the author's current clause-by-clause processing theory, that a unit corresponding to the basic clause is a stable and easily recognizable surface unit, and that it is also an important partial result and building block in the construction of a richer linguistic representation that encompasses syntax as well as semantics and discourse structure.

2.3 A Regular Expression for Basic Clauses.

Several versions of a regular expression for basic clauses were written by the author and preceded the one presented in Appendix 1, which was, applied to 60 files of Brown corpus tagged text of 2000 words each, newspaper texts A01-A20, scientific texts J01-J20 and fiction texts K01-K20.

The first half of the definition of *clause* introduces a few auxiliary definitions: comp for a set of complementizers, punct for a set of punctuation marks, and tense for a set of verb forms that are either certainly tensed ("BED" "BEDZ" "BEM" "BER" "BEZ" "DOD" "DOZ" "HVD" "HVZ" "MD" "VBD" "VBZ") or possibly tensed ("BE" "DO" "HV" "VB"). The definition of clause also uses the previously defined *brown-np-regex*. The second and larger part of the definition of *clause* consists of a union of six concatenations.

The first defines complete main clauses as consisting of a sequence of an optional coordinating conjunction CC followed by an obligatory basic noun phrase followed by optional non-clausal complements and an optional adverb followed by an obligatory tensed verb followed by anything expcept the punctuations or complementizers indicated in the list after (not ..., followed by optional punctuation.

The second defines clauses introduced by an obligatory CC followed by an optional adverb followed by an obligatory element which is either a tensed or participial verb form, followed by the same clause ending as in the first definition.

The third concatenation defines a subordinate clause as starting with an optional coordinating conjunction followed by an obligatory complementizer followed by the same clause ending as in the first and second definitions.

The remaining three definitions are of clause fragments rather than full clauses. Consider the following: The man (who liked ice cream,) ate too much.

In it, the relative clause makes a basic clause unit that breaks up the main clause into two clause fragments. The third concatenation defines noun phrase fragments that begin with a basic noun phrase followed optionally by one or more prepositional phrases, or sequences of CC np or $ np, followed by the same clause ending as in the other definitions. In the example above, (the man) would be a noun phrase fragment.

The fifth concatenation defines verb phrase fragments, e.g. (ate too much).

The sixth concatenation defines clause fragments that are adjuncts, i.e. adverbial phrases, prepositional phrases and adjective phrases. The typical case in which such a fragment is recognized is when it precedes another clause: (On a clear day,) (you can see forever).

2.4 Output of Regular Expression for Clauses.

The regular expression in Appendix 1 was automatically expanded into a deterministic fsa for clause recognition by Church's program. This rule compilation will not be described here. An excerpt from the result of applying it to the 60 files mentioned in the introduction to this section is presented in Appendix 2, where the location and nature of hand-corrections have been high-lighted. The hand-correction was guided by the following principles.

1) There should be at most one tensed verb per clause. This inserts a clause boundary after a tensed clause and before a tensed verb in the following kind of case, which the current regular expression matcher does not capture: (The announcement) (that the President was late) (was made late in the afternoon).

2) There should be a clause boundary after a sentence initial prepositional or adverbial phrase and before the sequence np tensed verb, whether or not they are separated by a comma: (At the summit in Iceland) (Gorbachev insisted ...).

3) There should be a clause boundary before CC followed by a tensed verb. Although the second concatenation in the clause regex aimes at capturing such clauses, it is not always successful in doing so because there is no way, given the current implementation of negation in the regular expression program, to state that a clause should end before a concatenation of items, i.e. before (* CC tense). Only single items can be negated at present. Example: (The Purchasing Departments are well operated) (and follow generally accepted practices).

4) There should be a clause boundary before a preposition (IN) followed by a wh-word, i.e. before (* IN (+ WDT WPO WP$ WRB WQL)). For the same reason given under 3), there is no way currently to state that a clause should end before such a sequence. Example: (The City Executive Committee deserves praise for the manner) (in which the election was conducted).

Several interesting observation were made in the course of doing these hand-corrections. For one, there were errors in the Brown corpus assignment of tags, in particular several errors confusing VBD and VBN, and there were errors where the sequence TO VB was tagged IN NN. More seriously, it turned out that the words "as" and "like", which have the property of functioning either like prepostions IN or subordinating conjunctions CS were always tagged CS, thus leading to incorrect recognition of clauses in many cases. Another problem for recognizing clauses on the basis of identifying tensed verbs was that the tag VB is applied to forms that are either infinitival or present tensed (or subjunctive), depending on context. It would have been better if such forms had been considered lexically ambiguous and given distinct tags. However, by and large the tagged Brown corpus is a very good and useful product, both in the choice of tags, and in the consistency with which they have been applied. Doing the

hand-correction also forced the realization that the clause recognition program, like the noun phrase recognition program, dependes crucially on accurate assignment of parts of speech to all words, on order to work well. For this task, Church's stochastic parts program is admirably suited, since it gives correct assignments in a very large number of cases, and it holds the potential of further improvement in its performance with further training.

2.5 Stochastic Recognition of Clauses.

As stated before, the regex *clause* was applied to sixty texts in the Brown corpus, and the output was hand-corrected. The hand-corrected files, containing an estimated total of at least 20,000 basic clauses, including clause fragments, were then used as training material for a stochastic recognition program. The training consisted of observing the location of clause opens and clause closes, and a special training specifically in locating tensed verbs. After training, the stochastic parts program and thereafter the stochastic clause recognizer was applied by K. Church to a large amount of Associated Press newswire text from May 26, 1987 (526 blocks, 2381353(8) bytes). An excerpt of the result is presented in Appendix 3. The result, again, is strikingly good.

A comparison of the nature and amount of errors in recognizing basic clauses in a sample of uncorrected regex output, and a sample of output from the stochastic clause program, can be made on the basis of Tables 1 and 2 at the end.

It appears that the stochastic program is more successful than the current regular expression method. However, certain improvements in the regex program could change that. What is needed is the facility to process generalized regular expressions, which admit the operations of complement and intersection, in addition to the operations of concatenation, union and Kleene star that characterize regular expressions. In any case there are some interesting differences in the kinds of errors made by the current regex program and the stochastic one for recognizing clauses. The regex program systematically errs by underrecognizing, never by overrecognizing, and in the selected portions that were scored, it only puts a few clause boundaries in the wrong place. It misses lots of clause boundaries, but the ones it gets are mostly correct.

The stochastic program, on the other hand, is able to get many clause boundaries correctly that elude the regular expression matcher, e.g. clauses not introduced by complementizers. The stochastic program errs both by overrecognizing and underrecognizing clauses, and sometimes it also places the clause open or clause close in the wrong place. Some cases of incorrect clause recognition are due to incorrect assignments of parts of speech to words. However, the total number of errors with the stochastic method (21) is smaller than the total number of errors with the regex method (40), for approximately the same number of clauses to be recognized, 304 versus 308. This is a very surprising outcome indeed, and if

taken literally, without any further weighting of the different types of errors, it means that the error rate for the stochastic method for recognizing clauses is 6.5%, as compared with 13% for the regex method.

3 On the Relation between Clauses and Intonation Units.

Finding basic clause units in arbitrary text is necessary in order to locate tonal minor phrases, which, in addition to a phrase accent, also have a boundary tone, and, particularly at slow rates of speech, a pause at the end of the phrase. The current experiment in text analysis has been concerned primarily with informative rather than imaginative prose, and envisages applications of the text-to-speech system to the reading of informative prose like newspaper text.

In the current Bell Labs text-to-speech system, tonal minor phrase boundaries are identified on the basis of commas, and tonal major phrase boundaries are identified on the basis of periods. Finding more tonal minor phrase boundaries by using syntactic structure, in addition to punctuation, is the problem I am trying to address with the methods described in this paper. In order to know where tonal minor phrase boundaries actually occur in the reading of informative texts, which typically have very long sentences (an average of 21 words compared with 14 words in general fiction based on Brown corpus data), it would be necessary to make recordings of several persons reading both authentic and prepared texts in a rhetorically explicit way, to borrow a phrase from Beckman & Pierrehumbert (1986), and then make extensive speech analyses of them, particularly of fundamental frequency movements and pauses. In the absence of such data for American English, the following kinds of boundaries between clauses and clause fragments were hypothesized to constitute intonation breaks with the status of tonal minor phrase boundaries. They are marked with # in the examples below.

a) After sentence initial adverbials and before np tense: (At the summit in Iceland) # (Gorbachev insisted ...)

b) After a relative clause and before a tensed verb: (A House Committee) (which heard his local option proposal) # (is expected) (to give it a favorable report).

c) After other noun phrases with clausal complements and before a tensed verb: (The announcement) (that the President was late) # (was made by the Press Secretary to the waiting journalists.)

d) Before a set of complementizers categorized CS in the Brown corpus, it is frequently the case that there is an intonation break: that/CS ..., whether/CS ..., if/CS ..., since/CS ..., because/CS ..., as/CS .... However, there are some exceptions to this , in particular:

(i) Comparatives: (This is not as/QL fast/JJ) (as/CS I would like ...) or (The theorem is more/RBR general/JJ) (than/CS what we have described)

(ii) The words as/CS and like/CS when used as prepositions, i.e. followed by noun phrases that are not subjects of clauses: (Jenkins left the White House in 1984,) (and joined Wedtech) (as/CS its director of marketing two years later.)

For testing purposes, short passages of seven consecutive sentences each from the Brown files, and four sentences each from the AP newswire stories were synthesized by the author, using the Bell Labs text-to-speech system. Those boundaries between clauses and clause fragments that are identified above were implemented in the same way that commas are, i.e. with a phrase accent belonging to the tonal minor phrase, final lengthening, a boundary tone, and a short pause of 200 ms. The results have not yet been subjected to perceptual tests.

There are some studies of the relation between clause units and intonation units that provide relevant data for future work. Gårding (1967) studied prosodic features in spontaneous and read Swedish speech. She found that in the spontaneous speech, pauses were equally divided between syntactic pauses and hesitation pauses, a syntactic pause being defined as one that coincides with a syntactic boundary. In the read speech, all pauses were syntactic pauses: "They appear between main clause and subordinate clause, before adverbial modifiers and between the different parts of an enumeration. The pause length is shortest in enumerations and before relative clauses (4-10 cs) and longest before adverbial modifiers and between complete sentences." (p. 48).

In a study of the intonational properties of relative clauses in British English, Taglicht (1977) compared the speech of a news broadcast with impromptu speech, and found that both genres separated nonrestrictive relative clauses prosodically. The news broadcast also separated a large proportion (71%) of the restrictive relative clauses prosodically.

A recent and very extensive study of the grammtical properties of intonation units, or tone units (TU) is Altenberg (1987). He studied a monologue of 48 minutes duration from the London-Lund Corpus of spoken English, and his results concerning the correlation of clause boundaries and tone unit boundaries are presented in Table 3 at the end.

4 Conclusion.

The study reported above shows that basic clauses, including basic noun phrases, are stable and surface recognizable units in the definitions they were given here, and that both finitary and stochastic methods can be used to find them in unrestricted text with a high degree of success. The comparison between the error rate of these two methods showed that the stochastic method performed better both in the recognition of basic noun phrases and basic clauses, which is an unexpected result.

REFERENCES

Allen, Jonathan, Hunnicutt, M. Sharon & Klatt, Dennis, 1987, From text to speech. The MITalk system, Cambridge, Cambridge University Press.

Altenberg, Bengt, 1987, Prosodic patterns in spoken English. Studies in the correlation between prosody and grammar for text-to-speech conversion. Lund Studies in English 76, Lund, Lund University Press.

Beckman, Mary & Pierrehumbert, Janet, 1986, Intonational structure in Japanese and English, Phonology Yearbook 3(1986), 255-309.

Church, Kenneth W., 1982, On memory limitations in natural language processing, Bloomington, Indiana, Indiana University Linguistics Club.

Church, Kenneth W., 1988, A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text, Proceedings of the Second Conference on Applied Natural Language Processing, Austin, Texas, ACL.

Ejerhed, Eva, 1987, Finding Noun Phrases and Clauses in Unrestricted Text: On the Use of Stochastic and Finitary Methods in Text Analysis (ms), AT& T Bell Labs.

Ejerhed, Eva & Church, Kenneth W., 1983, Finite State Parsing, in F. Karlsson (ed.), Papers from the Seventh Scandinavian Conference of Linguistics, University of Helsinki, Department of General Linguistics.

Francis, Nelson & Kucera, Henry, 1982, Frequency Analysis of English Usage, Lexicon and Grammar, Boston, Houghton Mifflin Company.

Gårding, Eva, 1967, Prosodiska drag i spontant och uppläst tal, in G. Holm (ed.), Svenskt talspråk, Stockholm, Almqvist & Wiksell, 40-85.

Jarvella, Robert, 1971, Syntactic Processing of Connected Speech, JVLVB 10, 409-416(1971).

Jarvella, Robert & Pisoni, D.B., 1970, The Relation between Syntactic and Perceptual Units in Speech Processing, JASA, 1970, 48, 84 (A).

Just, Marcel & Carpenter, Patricia, 1984, Using Eye Fixations to Study Reading Comprehension, in D. Kieras & M. Just (eds), New Methods in Reading Comprehension Research, Lawrence Earlbaum Associates, Hillsdale, New Jersey, 151-182.

Langendoen, D. Terrence, 1975, Finite-State Parsing of Phrase-Structure Languages and the Status of Readjustment Rules in Grammar, Linguistic Inquiry, Vol VI (1975), Number 4.

Liberman, Mark & Buchsbaum, Adam, 1985, Structure and Usage of Current Bell Labs Text to Speech Program, (ms), AT&T Bell Labs.

Taglicht, J., 1977, Relative clauses as postmodifiers: meaning syntax and intonation, in W.-D. Bald & R. Ilson (eds.), Studies in English usage, Frankfurt/M, Peter Lang, 73-107.

Wright,Charles, Bachenko, Joan & Fitzpatrick, Eileen, 1986, The contribution of parsing to prosodic phrasing in an experimental text-to-speech system, Proceedings of the 24th Annual Meeting of the Association for Computational Linguistics, Columbia University.

Table 1. Errors in regex recognition of clauses.

Regex Output

Story	Sentences	Clauses before	Clauses after	Under	Wrong-place
a01	28	86	104	18	1
j01	28	98	107	9	1
k01	28	87	97	10	1
Total	84	271	308	37	3

Table 2. Errors in stochastic recognition of clauses.

Stochastic Output

Story	Sentences	Clauses before	Clauses after	Under	Over	Wrong-place
STORY-1	15	64	64	0	1	1
STORY-2	15	52	51	0	1	0
STORY-3	15	45	46	1	0	2
STORY-4	30	141	143	8	4	3
Total	75	302	304	9	6	6

Table 3. The cooccurrence of clause boundaries and tone unit boundaries (from Altenberg 1987:57 Table 4:3).

Clause boundaries cooccurring with a TU boundary.

Clause boundary	Total	TU boundary	%
After initial clauses	29	29	100
Around medial clauses	15	15	100
Before finite adverbial clauses	46	46	100
Before adverbial ing-clauses	14	14	100
Before nonrestrictive relative clauses	26	26	100
Before asyndetic clause coordination	15	15	100
Around nonrestrictive appositive clauses	3	3	100
After postmodifying clauses	67	66	99
Before syndetic clause coordination	153	150	98
Before nonfinite postmodifying clauses	25	19	76
Before restrictive relative clauses	26	18	69
After comment clauses	13	9	69
Before adverbial infinitive clauses	12	8	67
Before comment clauses	13	8	62
Before nominal that-clauses	32	19	59
Before direct speech	7	4	57
Before nominal relative/interrogative clauses	16	7	44
Before nonfinite nominal clauses	21	7	33
Before clauses as prepositional complement	21	1	5

APPENDIX 1
Regular expression for basic clauses.

```
(defvar *clause*
 (let* ((comp '(+ "CS" "TO" "WDT" "WRB" "WPS" "WPO" "WP$" "WQL"))
     (punct '(+ "," "." "--" ":"))
     (tense '(+ "BE" "BED" "BEDZ" "BEM" "BER" "BEZ" "DO" "DOD"
                "DOZ" "HV" "HVD" "HVZ" "MD" "VB" "VBD" "VBZ")))
     `(+ (* (cl-user::opt "CC")   ;main clause: (CC) np tense ...
            ,*brown-np-regex*
            (cl-user::opt (++ (* (+ "CC" "IN" "$")
                                 ,*brown-np-regex*)))
            (cl-user::opt (+ "RB" "RBR"))
         ,tense
            (cl-user::opt (++ (not "," "." "--" ":"
                                   "CS" "TO" "WDT" "WRB" "WPS"
                                   "WPO" "WP$" "WQL")))
            (cl-user::opt ,punct))
         (* "CC"                 ;main clause: CC tense ...
            (cl-user::opt (+ "RB" "RBR"))
            (+ ,tense "VBG" "VBN" "BEG" "HVG")
            (cl-user::opt (++ (not "," "." "--" ":"
                                   "CS" "TO" "WDT" "WRB" "WPS"
                                   "WPO" "WP$" "WQL")))
            (cl-user::opt ,punct))
         (* (cl-user::opt "CC")        ;sub clause
            (++ ,comp)
            (cl-user::opt (++ (not "," "." "--" ":"
                                   "CS" "TO" "WDT" "WRB" "WPS"
                                   "WPO" "WP$" "WQL")))
            (cl-user::opt ,punct))
         (* (cl-user::opt "CC")        ;np clause fragment
            ,*brown-np-regex*
            (cl-user::opt (++ (* (+ "CC" "IN" "$")
                                 ,*brown-np-regex*)))
            (cl-user::opt (++ (not "," "." "--" ":"
                                   "CS" "TO" "WDT" "WRB" "WPS"
                                   "WPO" "WP$" "WQL")))
            (cl-user::opt ,punct))
         (* (+ ,tense "VBG" "VBN" "BEG" "HVG")
                                       ;vp clause fragment
            (cl-user::opt (++ (not "," "." "--" ":"
                                   "CS" "TO" "WDT" "WRB" "WPS"
                                   "WPO" "WP$" "WQL")))
            (cl-user::opt ,punct))
         (* (cl-user::opt "CC")        ;adjunt clause fragment
            (++ (* (+ "RB" "RBR" "RP" "QL" "*" "NR" "JJ" "JJR"
                      "IN" ,*brown-np-regex* )))
            (cl-user::opt (++ (not "," "." "--" ":"
                                   "CS" "TO" "WDT" "WRB" "WPS"
                                   "WPO" "WP$" "WQL")))
            (cl-user::opt ,punct)))))
```

APPENDIX 2

Sample of output of applying the regular expression *clause* as defined in Appendix 1, to Brown newspaper story A01. Hand-corrections are marked by double asterisks for under-recognized, and single asterisks for overrecognized clause boundaries.

[the/AT Fulton/NP-TL County/NN-TL Grand/JJ-TL Jury/NN-TL said/VBD Friday/NR ** an/AT investigation/NN of/IN Atlanta/NP 's/$ recent/JJ primary/NN election/NN produced/VBD no/AT evidence/NN] [that/CS any/DTI irregularities/NNS took/VBD place/NN./.]

[the/AT jury/NN further/RBR said/VBD in/IN term-end/NN presentments/NNS] [that/CS the/AT City/NN-TL Executive/JJ-TL Committee/NN-TL ,/,] [which/WDT had/HVD over-all/JJ charge/NN of/IN the/AT election/NN ,/,] [deserves/VBZ the/AT praise/NN and/CC thanks/NNS of/IN the/AT City/NN-TL of/IN-TL Atlanta/NP-TL for/IN the/AT manner/NN ** in/IN] * [which/WDT the/AT election/NN was/BEDZ conducted/VBN ./.]

[the/AT September-October/NP term/NN jury/NN had/HVD been/BEN charged/VBN by/IN Fulton/NN-TL Superior/JJ-TL Court/NN-TL Judge/NN-TL Durwood/NP Pye/NP] [to/TO investigate/VB reports/NNS of/IN possible/JJ irregularities/NNS in/IN the/AT hard-fought/JJ primary/NN] [which/WDT was/BEDZ won/VBN by/IN Mayor-nominate/NN-TL Ivan/NP Allen/NP Jr./NP ./.]

[only/RB a/AT relative/JJ handful/NN of/IN such/JJ reports/NNS was/BEDZ received/VBN ,/,] [the/AT jury/NN said/VBD ,/,]N [considering/IN the/AT widespread/JJ interest/JJ in/IN the/AT election/NN ,/,] [the/AT number/NN of/IN voters/NNS and/CC the/AT size/NN of/IN this/DT city/NN ./.]

[the/AT jury/NN said/VBD ** it/PPS did/DOD find/VB] [that/CS many/AP of/IN Georgia/NP 's/$ registration/NN and/CC election/NN laws/NNS are/BER outmoded/JJ or/CC inadequate/JJ and/CC often/RB ambiguous/JJ ./.]

[it/PPS recommended/VBD] [that/CS Fulton/NP legislators/NNS act/VB] [to/TO have/HV these/DTS laws/NNS studied/VBN and/CC revised/VBN to/IN the/AT end/NN of/IN modernizing/VBG and/CC improving/VBG them/PPO ./.]

[the/AT grand/JJ jury/NN commented/VBD on/IN a/AT number/NN of/IN other/AP topics/NNS ,/,] [among/IN them/PPO the/AT Atlanta/NP and/CC Fulton/NP-TL County/NN-TL purchasing/VBG departments/NNS ** which/WDT it/PPS said/VBD ** are/BER well/QL operated/VBN ** and/CC follow/VB generally/RB accepted/VBN practices/NNS] [which/WDT inure/VB to/IN the/AT best/JJT interest/NN of/IN both/ABX governments/NNS ./.]

APPENDIX 3

Sample of output of stochastic procedure for finding clause boundaries. Tensed verbs should be in bold face. In the recognition of these clauses, the constraint was enforced that there be at most one tensed verb per clause. Hand-corrections marked as in Appendix 2.

[former/AP U.S./NP Attorney/NN General/NN Ramsey/NP Clark/NP said/VBD! Monday/NR] [he/PPS believed/VBD!] [he/PPS had/HVD! found/VBN evidence/NN of/IN a/AT growing/VBG CIA/NP role/NN in/IN the/AT Philippines/NPS '/$ war/NN against/IN communist/NN rebels/NNS ./.]

[Clark/NP ,/,] [who/WPS arrived/VBD! last/AP week/NN] * [as/CS the/AT head/NN of/IN a/AT private/JJ ,/,] [human/JJ rights/NNS team/NN ,/,] [said/VBD!] [he/PPS hopes/VBZ!] [to/TO! document/VB the/AT evidence/NN] [and/CC present/VB it/PPO to/IN U.S./NP Secretary/NN of/IN State/NN George/NP P./NP Shultz/NP ./.]

[our/PP$ concern/NN is/VBZ! the/AT role/NN of/IN the/AT United/VBN States/NNS ,/,] [Clark/NP told/VBD! a/AT news/NN conference/NN ./.]

[we/PPSS believe/VB!] [we/PPSS can/MD! see/VB ,/,] [and/CC we hope/VB!] [to/TO! be/BE able/JJ] [to/TO! document/VB] * [before/CS we/PPSS are/BER!] * [through/RP in/IN our/PP$ report/NN ,/,] [evidence/NN clearly/RB establishing/VBG the/AT implementation/NN of/IN a/AT low-intensity/JJ campaign/NN here/RB ,/,] [with/IN violence/NN ,/,] [to/TO! kill/VB off/RP all/ABN opposition/NN ,/,] [every/AT opposition/NN to/IN authority/NN ,/,] [to/IN militarism/NN ./.]

[Ralph/NP McGehee/NP ,/,] [a/AT former/AP Central/JJ Intelligence/NN Agency/NN employee/NN ,/,] [said/VBD!] [he/PPS recognized/VBD! indications/NNS of/IN CIA/NP influence/NN in/IN the/AT Philippine/JJ military/NN 's/$ operations/NNS against/IN the/AT communist/JJ New/JJ People/NNS 's/$ Army/NN ./.]

[he/PPS cited/VBD! military/JJ search-and-destroy/JJ missions/NNS ,/,] [forced/VBN evacuation/NN of/IN civilians/NNS from/IN rebel-held/JJ areas/NNS and/CC the/AT increase/NN in/IN the/AT strength/NN of/IN civilian/JJ anti-communist/JJ vigilante/JJ groups/NNS ./.]

[the/AT allegations/NNS of/IN growing/VBG U.S./NP involvement/NN in/IN the/AT support/NN of/IN president/NN Corazon/NP Aquino/NP 's/$ government/NN came/VBD! with/IN claims/NNS by/IN Philippine/JJ leftists/NNS] [that/CS right-wing/JJ death/NN squads/NNS are/BER! operating/VBG freely/RB against/IN suspected/VBN leftists/NNS ./.]

Disambiguering i human oversættelse og i maskinoversættelse*

Frede Boje
Eurotra-DK

0. Indledning

Denne artikel udspringer af mit arbejde i Eurotra, hvor jeg især har beskæftiget mig med transfer mellem tysk og dansk, men de problemer, jeg omtaler med udgangspunkt i konkret materiale fra transfer-arbejdet, er temmelig generelle. Emnet ligger i grænseområdet mellem datalingvistik og leksikografi, men selv om jeg ikke kommer meget ind på datalingvistiske formalismer, har jeg forsøgt at anskue problemet fra datalingvistens synsvinkel snarere end fra leksikografens. Det betyder bl.a., at jeg ikke forudsætter fortrolighed med leksikografiske begreber.

Når jeg i det følgende bruger betegnelsen "ordbøger", betyder det almindelige tosprogede ordbøger til brug for mennesker, med mindre jeg udtrykkelig nævner andre former for ordbøger.

1. Aktiv / passiv ordbog

Det er i de senere år inden for tosprogs-leksikografien blevet almindeligt at skelne mellem aktive og passive ordbøger. Betegnelserne, der er indført af Smolik (1969)[1], er ikke særlig velvalgte, men de er vist efterhånden så fastgroede, at det er håbløst at forsøge at udskifte dem. Efter min mening ville det være bedre at tale om produktions- kontra receptionsordbøger eller - med Hausmann (1977) - Hinübersetzungs- kontra Herübersetzungswörterbücher.

Kort fortalt ligger forskellen i oversættelsesretningen i forhold til brugerens modersmål. Ved oversættelse til et fremmedsprog, altså Hinübersetzung, har man brug for en aktiv ordbog, ved oversættelse til sit modersmål, Herübersetzung, for en passiv ordbog. Der er således - principielt - behov for 4 ordbøger for et givet sprogpar, f. eks. mellem dansk og tysk:

1. dansk => tysk for danskere
2. dansk => tysk for tyskere
3. tysk => dansk for danskere
4. tysk => dansk for tyskere.

Hovedtanken i denne opsplitning er, at leksikografen kan - og bør - udnytte brugerens modersmåls-kompetence, når han udvælger de informationer, der skal medtages i ordbogen. Hvilke konkrete konsekvenser dette synspunkt får, vil jeg ikke uddybe her; det er beskrevet indgående i Kromann/Riiber/Rosbach (1984) og flere andre artikler af de samme forfattere. Lidt forenklet kan det siges således: Det er fremmedsprogets ordforråd, der kræver kommentarer (dvs. forklaringer, definitioner, eksempler osv.).

---

* Foredrag ved de Nordiske Datalingvistdage i København 3.-4. november 1987.

Det forudsættes altså, at brugeren ved Hinübersetzung umiddelbart forstår kildesproget, nemlig sit modersmål, mens han kan have behov for kommentarer af en eller anden slags for at kunne vælge den rigtige ækvivalent på målsproget. Omvendt kan han ved Herübersetzung have behov for kommentarer for at forstå den fremmedsprogede kildetekst, mens han forventes at beherske sit modersmål så godt, at han kan vælge den rette målsprogs-ækvivalent, når blot kildeteksten er forstået.

Denne skelnen gælder for ordbøger til mennesker. Ved maskin-oversættelse har hverken maskinen eller de programmer, man putter i den, nogen form for modersmåls-kompetence. Derfor har man brug for en ordbog, som så at sige er dobbelt aktiv:

	Kildesprog	Målsprog
Aktiv ordbog	- komm.	+ komm.
Passiv ordbog	+ komm.	- komm.
Maskin-ordbog	+ komm.	+ komm.

2. Hvor mange - og hvilke ækvivalenter?

2.1 Antallet af ækvivalenter

Den forskel, jeg hidtil har omtalt, er især principiel. I praksis har de fleste eksisterende ordbøger ikke klart definerede målgrupper og derfor ofte en blanding af kommentarer til lemmaerne, altså kildesprogsordene, og målsprogsækvivalenterne.

Derimod er der - både i teori og i praksis - en meget stor forskel på ordbøger til human- og maskinoversættelse m.h.t. hvad der er brug for af det, jeg i første omgang blot har kaldt "kommentarer". Denne forskel vil blive uddybet senere i artiklen, men inden da vil jeg redegøre for en række kvantitative aspekter, der danner baggrund for hovedproblemet.

For at begrænse undersøgelsernes omfang har jeg valgt at koncentrere mig om ord med begyndelsesbogstavet 'f', der i de forskellige værker optager fra knap 5% til godt 8% af pladsen. Eksempler, stikprøveundersøgelser, forklaringer osv. i det følgende er således hovedsagelig taget fra bøgernes afsnit med ord, der begynder med 'f'.

Først har jeg set på, hvor mange oversættelses-ækvivalenter der er pr. lemma. En undersøgelse af de første 500 F-ord i Gyldendals røde ordbog Tysk-Dansk (7) giver det resultat, der er vist i skemaets første dobbeltspalte[2].

Denne fordeling har jeg derefter sammenlignet med den samme ordbogs oversættelse af F-ordene i Heinz Oehlers "Grundwortschatz Deutsch"(15), der rummer de 2500 hyppigste ord på tysk.

	1 (500 første) F-ord uden hensyn til hyppighed		2 F-ord hørende til hyppigste ord		3 F-ord hørende til næsthyppigste ord		4 Ikke hørende til hyppigste eller næsthyppigste ord	
Antal ækvivalenter	Antal lemmaer	Procent	Antal lemmaer	Procent	Antal lemmaer	Procent	Antal lemmaer	Procent
1	340	68	10	25	26	29,9	335	69,5
2	99	19,8	5	12,5	19	21,8	97	20,1
3	27	5,4	4	10	10	11,5	27	5,6
4	15	3	5	12,5	7	8,1	14	2,9
5	8	1,6	1	2,5	9	10,3	6	1,2
6	3	0,6	3	7,5	4	4,6	3	0,6
7	3	0,6	3	7,5	2	2,3	-	
8	-		1	2,5	2	2,3	-	
9	3	0,6	2	5	2	2,3	-	
10	1	0,2	-		3	3,5	-	
11-15	-		2	5	3	3,5	-	
15	1	0,2	4	10	-		-	
I alt	500	100,0	40	100,0	87	100,1	482	99,9

Fordelingen af de 40 F-ord, der efter Oehlers angivelser hører til de 600 almindeligste tyske ord, er vist i skemaets 2. dobbeltspalte.

Resten af F-ordene i grundordforrådet (i alt 87 ord) havde den fordeling, der er vist i skemaets 3. dobbeltspalte.

Endelig har jeg sammenlignet med de ord fra første dobbeltspalte, der ikke hører til grundordforrådet. Deres fordeling er vist i skemaets sidste dobbeltspalte.

Det skal også nævnes, at det af forordet til ordbogen tydeligt fremgår, at man bevidst har "udeladt det meste af det ordforråd, der er umiddelbart forståeligt". Der er her i høj grad tale om lavfrekvente ord, der kun har én ækvivalent, f. eks. et stort antal uproblematiske fremmedord. Det vil sige, at procenten af ord med én ækvivalent snarere er højere, end optællingen i ordbogen viser.

Hvis vi generaliserer ud fra denne stikprøve - og det mener jeg man roligt kan tillade sig - kan vi se, at langt over halvdelen af det samlede ordforråd (i vores tilfælde 68%) ikke giver disambigueringsproblemer, fordi der kun er én oversættelsesækvivalent pr. lemma. Jo sjældnere et ord er, desto større er sandsynligheden for, at det kun har én ækvivalent, og omvendt: jo almindeligere et ord er, desto flere ækvivalenter har det som regel, og desto sværere er det derfor at disambiguere sikkert.

Lingvistisk og leksikografisk er det altså sværere at få et maskinoversættelsessystem til at oversætte de 5-600 almindeligste ord rigtigt i en rimelig procentdel af tilfældene end at udvide et velfungerende systems ordforråd fra f. eks. 2.000 til 20.000 ord eller for den sags skyld til 100.000 ord. Den sidste store udvidelse er ikke et lingvistisk problem, men et datamatisk, primært et kapacitetsproblem.

2.2 Hvilke ækvivalenter skal vælges?

2.2.1 Sandsynlighed

Foreløbig har vi kun set på antallet af ækvivalenter, som en almindelig ordbog "tilbyder". Når man skal tage stilling til, hvilke ækvivalenter der skal bruges i et maskinoversættelsessystem, kan man med fordel inddrage en anden form for statistik, nemlig sandsynlighedberegning. Når et ord i ordbogen f. eks. har et stort antal ækvivalenter, er det rimeligt at undersøge, om antallet kan reduceres til noget mere overkommeligt. Man kan bl.a. overveje, hvor sandsynlig en oversættelsesmulighed er.
Lad os f. eks. se, hvad der står i ordbogen (7) om "fahrbar":

Fähnlein *n* -. trop
Fähnrich *m* -e. officersaspirant
Fahr-abteilung *f* (mil.) motoriseret afdeling; **-ausweis** *m* rejsehjemmel; *SZ* kørekort; **-bahn** *f* kørebane; **-bahnmarkierung** *f* kørebaneafstribning; **-bahnrand** *m* rabat;
→ **-bar** *adj* transportabel, som kan køres; (gld) farbar, fremkommelig; *-er Tisch* rullebord; **-bereich** *m* aktionsområde; **-bereit** klar til afgang/start, køreklar

De 4 ækvivalenter hører betydningsmæssigt sammen to og to, hvilket er angivet ved semikolon. Den betydning, som udtrykkes med ækvivalent 3 og 4 er markeret som (gld.), altså gammeldags, en vurdering, som deles af ordbøger som Wahrig (11), DudGW (13) og DUW (14). Vi kan altså med god samvittighed se helt bort fra disse to ækvivalenter; tilbage bliver 1 og 2, som jeg ikke vil kommentere yderligere i denne omgang.

Desværre er det de færreste tilfælde, hvor man har så enkle metoder til at afgøre, hvad der kan skæres fra. Normalt må skønnet baseres på ens erfaring og common sense. Et tilstrækkelig stort relevant korpus ville være en stor hjælp. Det korpus, vi arbejder med i Eurotra i øjeblikket, er, dels p.g.a. sit ringe omfang, dels p.g.a. sin tilblivelse, ikke egnet til at give tilstrækkelig sikre kriterier.

2.2.2 Generalisering

Det er ikke kun hyppigheden, der spiller en rolle. En væsentlig faktor er også den enkelte ækvivalents betydningsomfang. Ofte har man i ækvivalentrækken en generel glose, der kan bruges i alle eller næsten alle tilfælde, plus nogle mere eller mindre synonyme udtryk, der hver for sig har en mere begrænset anvendelse. Her vil man normalt vælge det mest generelle udtryk, selv om man derved går glip af nogle nuancer.

Et eksempel herpå er oversættelsen af det danske "bestanddel", som ifølge Vinterberg & Bodelsen (1) har fire engelske oversættelses-ækvivalenter:

1. component
2. constituent
3. element
4. ingredient

Ifølge vores engelske samarbejdspartnere (der jo har ansvaret for oversættelserne til engelsk) vil "component" altid kunne bruges, mens de tre andre forslag har hver deres begrænsning. Man kan altså nøjes med at bruge én oversættelse: "bestanddel" => "component".

2.2.3 Tilføjelser

Selv om vi i Eurotra-projektet i øjeblikket af praktiske grunde prøver at begrænse antallet af ækvivalenter mest muligt, kan vi også blive nødt til at tilføje oversættelsesmuligheder, som ikke står i de gængse ordbøger. Det hænger sammen med, at vi i den nuværende fase arbejder korpus-baseret. De ord, der står i de tilsvarende korpora på de andre sprog, skal mindst kunne oversættes i den eller de betydninger, der optræder i disse korpora, og det giver undertiden oversættelser, der ikke findes i ordbøgerne. Et ekstremt eksempel her - som ikke er med F - er verbet "ansprechen", som kun optræder én gang i det tyske korpus. Det har i konteksten en betydning, som bedst gengives med "vedrøre". Den tysk-danske ordbog har i alt 25 oversættelsesforslag, men ikke "vedrøre" eller dermed beslægtede udtryk.

3. Monolingval/bilingval/multilingval disambiguering - Readings

Når man i lingvistisk litteratur har beskæftiget sig med ambiguitet og disambiguering, har det - såvidt jeg har kunnet se - normalt været i et monolingvalt perspektiv, og det er mit indtryk, at mange datalingvister regner med, at en monolingval disambiguering er tilstrækkelig også i oversættelsessammenhæng.

Tager vi en tekststreng - med F - som:

<fortaler>,

vil mange måske mene, at opgaven er løst, når man har fundet ud af, om det drejer sig om en form af:

1. "fortaler", cat=n, f. eks.: "Han er fortaler for en ny politik".
2. "fortale", cat=n, f. eks.: "Han har skrevet fortaler til begge bøger" eller
3. "fortale sig", cat=v, f. eks.: "Han fortaler sig let, når han er nervøs".

I oversættelsessammenhæng er disambigueringen imidlertid kun færdig, hvis det viser sig, at hvert af disse tre udtryk har lige præcis én oversættelsesækvivalent. Ved oversættelse til tysk eller engelsk er det tæt på at være rigtigt, men lad os tage en anden sammensætning med "-tale", nemlig:

<aftale>

Her vil man ud fra et rent dansk synspunkt mene, at det må være tilstrækkeligt at skelne mellem verbet og substantivet, for "en aftale er en aftale". I hvert fald er der hverken i NDO (9), ODS (10) eller Dansk Sprogbrug (11) så meget som en antydning af en skelnen mellem flere betydninger af substantivet.

Ikke desto mindre får man ved oversættelse til engelsk og tysk problemet med at vælge mellem ækvivalenterne[3]:

engelsk:	tysk:
1. agreement	1. Abkommen
2. appointment	2. Absprache
3. arrangement	3. Rücksprache
4. collusion	4. Termin
5. date	5. Verabredung
	6. Vereinbarung
	7. Vertrag

Denne liste kan muligvis reduceres med et par ækvivalenter efter de principper, jeg tidligere har nævnt, men tilbage bliver den ubehagelige kendsgerning, at f. eks. i forbindelse med oversættelse til tysk er substantivet "aftale" først disambigueret, når der kan skelnes mellem 5-7 readings af ordet.

Reading

Begrebet "reading", som jeg nu brugte, er desværre også problematisk, ikke kun fordi vi savner et godt dansk ord for det. I de fleste tilfælde svarer det meget godt til ordet "betydning", men langtfra altid. F. eks. virker det i vores eksempel lidt mærkeligt at sige, at substantivet "aftale" alt efter synsvinkel har én, fem eller syv betydninger.

Det, der især gør "reading" til et problematisk begreb, er, at mange bruger ordet, som om det var en nøje defineret størrelse, skønt der lægges forskellige betydninger i ordet. Det samme gælder "lexical unit", forkortet LU. F. eks. er afgrænsningen mellem, hvad der er to LU'er og hvad der er to "readings" af én LU, ikke særlig klar.

Jeg vil gerne sætte de to begreber i relation til begreberne "homonymi" og "polysemi", men først lige afklare, hvilke betegnelser jeg bruger for nogle fænomener, der selv er simple, men hvor terminologien tilsyneladende ikke er fast, i hvert fald ikke på dansk. For de tre engelske betegnelser:

feature: attribute = value (a feature is an a/v pair)

bruges:

feature: træk = værdi.

eks: number = plural

Som bekendt er det vanskeligt at give sikre kriterier for en skelnen mellem homonymi (her som homografi) og polysemi, og der har været argumenteret for, at denne skelnen inden for datalingvistikken skulle være irrelevant, at der f. eks. ikke skulle være grund til at lægge mere vægt på ordklasse end på alle andre træk i en feature-beskrivelse.

I én forstand er det rigtigt: I en unificeringsgrammatik er det ligegyldigt, om de værdier, der i en given regel skal (eller ikke må) matche, er værdier for ordklasse eller f. eks. for tællelighed, komparation eller tempus.

Det er muligt, at man i en analyse, der kun skal bruges monolingvalt, ikke har behov for en skelnen, men til oversættelsesformål mener jeg, at det er i det mindste praktisk, og måske nødvendigt at opstille nogle klare operationelle kriterier for en skelnen, der ikke nødvendigvis falder sammen med en grænsedragning efter f. eks. etymologiske principper. Det kan også være relevant at operere med forskellige kriterier for forskellige analyseniveauer. Mit forslag til skelnen lyder således:

Ord, der (i deres opslagsform) skrives ens, men tilhører forskellige ordklasser, er homografer og dermed forskellige LU'er. Er det substantiver, anses de for homografer - og altså forskellige LU'er, hvis de har forskelligt køn. Bortset herfra gælder, at hvis de tilhører samme ordklasse, men har forskellige værdier for mindst ét træk, er de polysemer og dermed forskellige readings af samme LU.

Denne opdeling er foretaget til praktiske formål og kan selvfølgelig forfines, hvis man føler behov for det. Det vil således nok være naturligt at inddrage i hvert fald bøjning som yderligere kriterium. Intuitivt virker det f. eks. mærkeligt at behandle 'die Bank' (= bank) og 'die Bank' (= bænk) som readings af samme LU[4].

4. Hvordan disambiguerer man?

Der skal ikke her siges så meget om, hvor disambigueringen finder sted; det uddybes i Annelise Bech og Poul Andersens artikel i dette nummer af Lambda[5]. I stedet vil jeg koncentrere mig om, hvilke midler man har.

Man kan se disambigueringsprocessen som en form for regelmatchning: Hvis en tekstenhed opfylder nogle bestemte betingelser, udløses et bestemt valg mellem ækvivalenter. Disse betingelser kan være angivet i en ordbog, en grammatik eller en anden form for opslagsværk, eller de findes inde i oversætterens hoved, som viden eller intuition. Det sidste tilfælde unddrager sig beskrivelse, så vi holder os til elektroniske og trykte medier.

For at der kan finde en matchning sted, skal det altså for det første være muligt at opstille nogle klare betingelser, for det andet skal det være muligt at konstatere, hvornår betingelserne er opfyldt. Det lyder banalt og selvindlysende, men det er alt andet end simpelt at anvende principperne i praksis. Ved maskinoversættelse er det meget sværere at opfylde krav nr. 2 end ved human-oversættelse, og det får også indvirkning på, hvilke betingelser der kan opstilles.

Human-oversætteren kan inddrage sin viden om verden og sin forståelse at teksthelhed, situationskontekst m.m. og kan drage

alle mulige analogislutninger, når han skal afgøre, om betingelserne for et givet valg er opfyldt.

Ved maskinoversættelse kan der stort set kun bruges betingelser, som går på forekomsten af bestemte fænomener i teksten. Mere konkret drejer det sig om tilstedeværelsen af størrelser med bestemte features eller bestemte værdier på trækkene.

Et afgørende spørgsmål er nu: Finder man i ordbøgerne oplysninger af denne type?

4.1 Disambigueringskriterier i ordbøgerne

Et fællestræk ved de fleste ordbøger, både mono- og bilingvale, er, at de vigtigste og undertiden næsten eneste midler til disambiguering er forklaringer og eksempler. "Forklaringer" er her brugt som overbegreb for definitioner, parafraser, synonymer osv. Uanset hvor relevante og velvalgte disse informationer måtte være set fra et almindeligt leksikografisk synspunkt, er de totalt ubrugelige til maskinoversættelse, vel at mærke direkte. Maskinen kan ikke stille noget som helst op med en oplysning i ordbogen om, at et givet ord f. eks. er synonymt med et andet. Derimod kan gode forklaringer og eksempler selvfølgelig være en hjælp for de mennesker, der laver ordbøgerne til maskinen.

4.1.1 Bilingvale ordbøger

De bilingvale ordbøger, oversættelsesordbøgerne, indeholder normalt mindst én oplysning, der direkte kan angives som et feature, nemlig ordklassen; for substantivers vedkommende angives tit også køn, for verbernes vedkommende transitivitet.

Desuden er der én type forekomst-relation, der ofte angives, nemlig den relation, der på engelsk kaldes "co-occurrence", men som vist ikke er navngivet på dansk. Det drejer sig typisk om forholdet mellem adjektiv og substantiv, altså den kendsgerning, at oversættelsen af et adjektiv ofte afhænger af, hvilket substantiv det modificerer. F. eks. kan man se i den dansk-tyske ordbog, at "høj" hedder "hoch" om et hus og "groß" om et menneske. Det store problem ved den slags oplysninger er, at det som regel ikke klart fremgår, hvilken form for co-occurrence der er tale om, altså om en given oversættelse af adjektivet er knyttet til ét bestemt substantiv, eller om substantivet står som repræsentant for en klasse - og i givet fald hvorledes denne klasse er afgrænset.

Selv hvis der er gjort forsøg på en afgrænsning, er den ofte vag og skal tages med forbehold. I eksemplet med "høj" står der således:

> 3. (om mennesker) groß (1,80 g.),

Ikke desto mindre finder man under "hoch" eksemplet

> "ein hoher Beamter",

og embedsmænd er dog normalt også en slags mennesker. Længere nede i artiklen står der:

5. (stofpåvirket) high [hai].

Det er normalt også kun relevant i forbindelse med mennesker.

Den slags inkonsekvenser spiller tit ikke nogen særlig rolle for den hærdede ordbogsbruger, fordi han er vant til at skulle hente supplerende disambiguerings-kriterier i eksemplerne og i øvrigt bruge sin sunde fornuft, men det gør det vanskeligt at omsætte betingelserne til features, som kan bruges i en maskine.

Ud over de ovennævnte oplysninger er der i de fleste oversættelsesordbøger næsten ingen informationer, der kan bruges som kriterier ved ækvivalentvalget, når de skal kunne udtrykkes ved feature-beskrivelser. Med andre ord: Oversættelsesordbøger kan bruges som en hjælp til at finde oversættelsesækvivalenter, men kun i ringe omfang til ud fra formelle kriterier at afgøre valget mellem dem, altså til at disambiguere maskinelt.

4.1.2 Monolingvale ordbøger

Ser vi nu på de monolingvale ordbøger, kan vi konstatere, at der er ret stor forskel på, hvilke informationer de giver, og hvilken form informationerne har. Som i de bilingvale ordbøger indtager eksemplerne en fremtrædende plads, men i de monolingvale ordbøger er forklaringsdelen som regel mere omfattende og rummer ofte egentlige definitioner.

Forskellen mellem ordbøgerne ligger især i, hvilke formaliserede eller direkte formaliserbare informationer de giver.

Næsten alle har oplysning om ordklasse, om substantivers køn (selvfølgelig ikke på engelsk) og mere eller mindre udtømmende oplysninger om bøjning[6]. Men derudover har nogle ordbøger systematiske oplysninger om opslagsordets omgivelser. Det gælder først og fremmest to ordbøger, Longmans Dictionary of Contemporary English (16) og dtv-Wörterbuch der deutschen Sprache (12). Longman har et meget udbygget kodesystem med bogstaver og tal, der angiver kombinationer af bøjningstype og distribution, mens dtv-Wörterbuch udelukkende bruger talkoder til at udtrykke lignende informationer.

Når man undersøger, hvorledes man kan udnytte disse oplysninger ved maskinoversættelse, konstaterer man, at de - med undtagelse af en vigtig gruppe informationer - som regel ikke er til særlig stor nytte. De er så sproginterne, at de er absolut relevante, når det drejer sig om tekstproduktion på det pågældende sprog, her altså henholdsvis engelsk og tysk, men de udgør sjældent kriterier ved disambiguering.

Undtagelsen er verberne. Her er der efterhånden udviklet metoder til en ret detaljeret beskrivelse af verbernes valens, og det ser ud til, at disse valensbeskrivelser kan gøre stor nytte ved udformningen af oversættelseskriterier.

4.2 Monolingvale readings og oversættelsesækvivalenter

I monolingvale ordbøger foretages der en inddeling i readings, der normalt begrundes i forskelle i ordenes betydning, og i oversættelsesordbøger er den væsentligste grund til at angive flere ækvivalenter, at de udtrykker forskellige betydninger. Det ville derfor være nærliggende at antage, at den skelnen mellem betydninger, der foretages monolingvalt, kunne bruges til at vælge mellem forskellige oversættelsesmuligheder.

Det kan desværre ikke uden videre lade sig gøre. Der er en række problemer forbundet med det.

(1) Det første er det tekniske problem, jeg allerede har nævnt, at forskellen mellem readings ofte kun angives ved forklaringer og eksempler. Det vil f. eks. være svært at formalisere forskellen på de første to readings under substantivet "fault" i Longman:

> **1** a mistake or imperfection: There are several faults in that page of figures.| a small electrical fault in the motor
>
> **2** a bad point, but not of a serious moral kind, in someone's character: Your only fault is that you won't do what you're told.| I love her for her faults as well as for her virtues.

(2) Det næste problem er, at "betydning" og "betydningsforskel" er så subjektivt bestemte begreber, at det er svært at blive enige om, hvor mange betydninger et ord har, og hvorledes forholdet er mellem disse betydninger. Det ses tydeligt, hvis man sammenligner beskrivelsen af det samme ord i forskellige monolingvale ordbøger. Ikke blot er antallet af readings og afgrænsningen mellem dem forskellige, men undertiden ser man også, at et eksempel, der i én bog skal illustrere én betydning, er identisk med eller svarer nøje til et eksempel, der i en anden bog illustrerer en anden betydning.

Hvis man ved oversættelsen vil tage udgangspunkt i en monolingval disambiguering i kildesproget, må man derfor tage stilling til, hvilke kriterier der skal lægges til grund og altså bl.a., om man vil gå ud fra en enkelt ordbog, eller om man vil gå eklektisk til værks. For behandlingen af dansk som kildesprog stiller sagen sig lidt anderledes. Vi har ikke noget at vælge imellem, da der ikke findes noget værk på dansk, der svarer til de andre sprogs Longman, Wahrig, Petit Robert osv., men vi må i stedet finde ud af, hvordan de sparsomme oplysninger i NDO skal suppleres.

(3) Det tredie problem er hovedproblemet. Det hænger nært sammen med det foregående, og det er oven i købet ekstra kompliceret i Eurotra-sammenhæng, fordi projektet ikke er bilingvalt, men multilingvalt.

Problemet er, at det ville være lettere at lave simpel transfer, hvis kildesprogsanalysen leverede lige præcis de readings, der udløser forskellige oversættelser, men samtidig ved vi, at man ved oversættelse til forskellige sprog har brug for forskellige betydningsafgrænsninger. Desværre er der mig bekendt ikke nogen, der kan sige noget særlig konkret om dette sidste fænomen.

For at belyse problemet kan vi se på Longman-eksemplet ovenfor. Vi kan på den ene side konstatere, at det fra en dansk synsvinkel er ligegyldigt, om man kan skelne mellem reading 1 og 2, for de skal begge oversættes ved "fejl". På den anden side kan vi ikke vide, om de måske skal oversættes forskelligt til f. eks. fransk eller græsk, så det alligevel kunne være relevant at kunne redegøre for forskellen.

Lad os se på et dansk eksempel: Verbet "føre" har i NDO 4 readings. Nr. 2 er forklaret som "stå i spidsen for" og rummer følgende eksempler, som jeg har nummereret:

1. føre en hær
2. Niels førte (ɔ: var forrest) under slutspurten
3. føre bog over sine udgifter
4. føre hus
5. føre krig
6. føre en sag
7. de førende (ɔ: toneangivende) kredse
8. føre ordet
9. føre en samtale
10. føre et rædsomt sprog
11. det er et skrækkeligt liv, han fører
12. refleksivt: hun forstår at føre sig ɔ: optræder smukt.

Man kan godt spørge sig selv, om "stå i spidsen for" er en rimelig parafrase for verbet i de anførte eksempler - jeg ved f. eks. ikke, hvordan man kan "stå i spidsen for et rædsomt sprog". Sagt på en anden måde: Det er et spørgsmål, hvor meget betydningsfællesskab verbet har i disse eksempler. Men lad os nu se på, hvad der sker ved oversættelse. Jeg har ikke fået 'native speakers' til at checke det følgende, men hvis jeg har brugt ordbøgerne rigtigt, og hvis man kan stole på dem (?!), skal (eller kan) "føre" i de anførte eksempler oversættes således:

	Engelsk:	Fransk:	Italiensk:	Tysk:
1.	command	conduire	?	führen
2.	lead	mener le peloton	essere in testa	führen
3.	keep	faire (registre)	tenere	führen (Haushalt)
4.	keep	tenir	governare	führen
5.	make	faire	fare	führen
6.	conduct	plaider	difendere	führen
7.	leading	prédominant	predominante	führend
8.	act as spokesman	porter	essere portavoce	führen
9.	carry on	soutenir	?	führen
10.	use	tenir	?	führen
11.	lead	trainer	condurre	führen
12.	carry oneself	se conduire	avere un bel portamento	sich führen

Oversættelsen til tysk er her ret simpel - jeg kan endda tilføje, at også samtlige eksempler under reading 1 og 3 naturligst oversættes med "führen". Hvis man omvendt prøver at oversætte de eksempler med "führen", der står i Wahrig, får man til gengæld brug for en halv snes danske ækvivalenter.

Bortset fra det tyske er det tydeligt, at eksemplerne skal oversættes så forskelligt, at der ikke er vundet noget som helst ved at lave en betydningsgruppering af det danske verbum.

Nu kunne man tænke sig, at verbet "føre" var et særlig ondskabsfuldt eksempel, og det er rigtigt, at det er værre end gennemsnittet, men det er ikke ekstremt med hensyn til antal ækvivalenter. Til gengæld er det meget almindeligt forekommende, så vi er nødt til at kunne gøre noget ved det. Det tyske "führen" hører til de 600 hyppigste ord, og jeg vil tro, at hyppigheden af "føre" er nogenlunde den samme.

Nogle vil måske mene, at ovenstående eksempel især viser, at der er behov for at få gjort noget ved kollokationer. Det er også rigtigt.

Det generelle spørgsmål, som sættes i relief af eksemplet med "führen", er, hvor stor overensstemmelse der er mellem de readings, man finder frem til ved den monolingvale analyse, og de oversættelses-ækvivalenter, der skal bruges.

Endnu er vores erfaringsmateriale ikke så stort, at vi kan sige ret meget om det, men et eksperiment, som vi har udført sammen

med den tyske Eurotra-gruppe, tyder på, at en opdeling af de enkelte verber efter valens er en stor hjælp i transfer-arbejdet, primært efter syntaktisk valens, men også efter semantiske restriktioner. I de tilfælde, hvor syntaktiske oplysninger ikke var tilstrækkelige, var der dog en tendens til, at oversættelseskriterierne faldt i to grupper: enten var trækket +/- HUM(AN) afgørende, eller også måtte man opfinde ad hoc-regler. Eksperimentet er beskrevet i Boje & al. (1986).

5. Konklusion

I abstract'et til dette foredrag opstillede jeg en tese. Jeg vil slutte med at gentage tesen i udbygget og kommenteret form.

Jeg hævdede, at "en væsentlig del af de oplysninger, der er relevante" for maskinoversættelse, "enten ikke findes i de almindelige ordbøger eller højst er implicitte", og jeg mener at have vist, at det gælder fuldt ud for oversættelses-ordbøgerne. I nogle monolingvale ordbøger findes en række formaliserede oplysninger; det endnu uafklarede spørgsmål er, i hvor høj grad de er relevante for oversættelse.

Hvorledes "maskinoversættelse stiller særlige krav til oplysningernes formaliserbarhed", mener jeg også at have vist, selv om det næppe har været nogen overraskelse for ret mange.

Endelig siger jeg i abstractet, at "formaliseringen ... med fordel vil kunne udnyttes i fremtidige ordbøger til human-oversættelse". Den slags påstande er svære at dokumentere, især så længe der er så få resultater at fremlægge, men lige så vel som den øgede formalisering har gjort monolingvale ordbøger bedre, er der for mig ingen tvivl om, at det samme vil kunne ske med oversættelsesordbøgerne, efterhånden som resultaterne af formaliseringsarbejdet viser sig.

Jeg har ikke givet svaret på to vigtige spørgsmål, som ikke står i abstractet, men måske ligger der - implicit:

(1) Hvordan konkretiserer og formaliserer man implicitte oplysninger?

(2) Hvordan fremtryller og formaliserer man de oplysninger, som er nødvendige for fuldstændig disambiguering, men som ikke findes i nogen opslagsværker?

Der kan endnu kun gives meget ufuldkomne svar på disse to spørgsmål. Det arbejde, der i øjeblikket udføres i Eurotra, omfatter bl.a. forsøg på at finde frem til et fælles system til kodning af semantiske træk. Forhåbentlig vil erfaringerne fra dette arbejde snart kunne udmøntes i en redegørelse, der rummer i hvert fald nogle af svarene på disse to spørgsmål.

Noter

1. Henvisning til ordbøger sker ved et tal i parentes. Dette tal angiver værkets nummer i litteraturlisten. "Anden citeret litteratur" angives på traditionel vis (som her): forfatter + årstal.

2. Optællingen skal selvfølgelig tages med et vist forbehold, ikke kun fordi det er en stikprøve, men især fordi kriterierne for optællingen altid kan diskuteres.
a. Der er kun medregnet ord og komplekse udtryk, der er anført som danske ækvivalenter til lemmaet alene, mens komplekse tyske udtryk, som indeholder lemmaet, men ikke er oversat kompositionelt, er holdt uden for beregningerne. Det gælder altså bl.a. idiomatiske udtryk.
b. Der er ikke gjort forsøg på at vurdere, hvilke ækvivalenter der er helt eller delvis synonyme. Til gengæld er ord (eller mere korrekt: tekststrenge), der er anført flere gange som ækvivalent for samme lemma, kun talt med én gang, uanset at de (i hvert fald efter ordbogsforfatterens mening) må anses for polysemer eller homografer. Antallet af ækvivalenter er derfor ikke automatisk lig med antallet af "betydninger".

Endelig vil antallet af ækvivalenter selvfølgelig også afhænge af bl.a. ordbogens omfang. Sammenligner man f. eks. de to engelsk-danske ordbøger fra Gyldendal, er mængden af ækvivalenter i den røde ordbog (3) (i hvert fald normalt) en ægte delmængde af ækvivalenterne i Kjærulff Nielsens store ordbog (2).

Eksempel:
fahrbar (adj): transportabel, som kan køres; (gld) farbar, fremkommelig; -er Tisch: rullebord

er talt som 4 ækvivalenter, "(fahrbar)-er Tisch" er ikke medregnet.

3. Fundet i hhv. (1) og (7).

4. På den anden side er forholdet mellem bøjning, syntaks og betydning ikke altid så enkelt. F. eks. er der i velplejet tysk en syntaktisk og betydningsmæssig forskel på det stærkt og det svagt bøjede verbum 'hängen' ('er hängte' er transitivt, 'er hing' er intransitivt), mens en tilsvarende skelnen på dansk er næsten opgivet. De fleste danskere bruger 'hængte' og 'hang' i flæng, både transitivt og intransitivt.

5. De to foredrag er udarbejdet uafhængigt af hinanden, og ingen af os udtaler sig på Eurotras vegne.

6. Der er her tale om et forsømt område inden for leksikografien, se f. eks. Kromann (1985).

LITTERATURLISTE

1. Bilingvale ordbøger

(1) **Dansk-engelsk Ordbog** v. H. Vinterberg og C.A. Bodelsen, 2.udg. 7. opl. (1985)

(2) **Engelsk-dansk Ordbog** v. B. Kjærulff Nielsen, 2. udg. 3. opl. (1985)

(3) **Engelsk-dansk Ordbog** af Jens Axelsen (Gyldendals røde ordbøger) 10. udg., 7. opl. (1985)

(4) **Dansk-fransk Ordbog** v. A. Blinkenberg og P. Høybye, 3. udg. (1976)

(5) **Dansk-Italiensk Ordbog** v. P. Høybye og J. Mengel, 2. udg., 2. opl. (1979)

(6) **Dansk-tysk Ordbog** v. E. Bork (Gyldendals røde ordbøger) 8. udg. (1980)
(7) **Tysk-dansk Ordbog** v. E. Bork (-"-) 11. udg. (1982)

2. Monolingvale ordbøger

2.1. Danske

(8) **Dansk Sprogbrug** v. E. Bruun (-"-) 1. udg. (1978)

(9) **Nudansk Ordbog** v. E. Oxenvad 11. udg. (1982) (NDO)

(10) **Ordbog over det dansk Sprog** Bd. 1-28 (1919-56) (ODS)

2.2. Udenlandske

(11) **Deutsches Wörterbuch** v. G. Wahrig, Neuausgabe (1980) (Wahrig)

(12) **dtv-Wörterbuch der deutschen Sprache** v. G. Wahrig (1978) (dtv-Wb)

(13) **Duden: Das große Wörterbuch der deutschen Sprache in sechs Bänden** (1976-81) (DudGW)

(14) **Duden: Deutsches Universalwörterbuch** (1983) (DUW)

(15) **Grundwortschatz Deutsch** v. H. Oehler (1966)
(frekvensoplysningerne taget fra den danske bearbejdelse:
Tysk-dansk Grundordbog v. O. Børløs Jensen (1970))

---
(16) **Longman Dictionary of Contemporary English** (1985) (Longman)

(17) **Petit Robert 1:** Dictionnaire de la Langue Francaise (1986)

(Der er ikke lagt vægt på at give fuldstændige bibliografiske oplysninger; kun de vigtigste data er medtaget. Udgaverne er de faktisk anvendte, selv om der i nogle tilfælde findes nyere udgaver)

3. Anden citeret litteratur:

Boje & al. 1986: Frede Boje, Birgit Weck and Hanne Ruus: The Choice of German and Danish Target LUs, based on Governor and Complement Information. (Internt Eurotra-papir, maj 1986)

Hausmann 1977: Franz Josef Hausmann: Einführung in die Benutzung der neufranzösischen Wörterbücher. Tübingen 1977. (Romanistische Arbeitshefte 19).

Kromann 1985: H.-P Kromann: Zur Selektion und Darbietung syntaktischer Informationen in einsprachigen Wörterbüchern des Deutschen aus der Sicht ausländischer Benutzer. In: Lexikographie und Grammatik. Akten des Essener Kolloquiums zur Grammatik im Wörterbuch 28.-30.6.1984. Hrsg. H.Bergenholtz & J. Mugdan. Tübingen 1985.

Kromann/Riiber/Rosbach 1984: H.-P. Kromann/Th. Riiber/P. Rosbach: "Active" and "Passive" Bilingual Dictionaries: The Scerba Concept Reconsidered. In: R.R.K. Hartmann (ed.): LEXeter '83 Proceedings Vol. II. Bilingual Lexicography and the Learner's Dictionary. Tübingen 1984.

Smolik 1969: W. Smolik: "Aktives" Wörterbuch Deutsch-Russich. In: Nachrichten für Sprachmittler H.3. 1969, 11-13.

A strategy for solving translation relevant ambiguities in a multi-lingual machine translation system.

Poul Andersen and Annelise Bech

EUROTRA-DK
Njalsgade 80,
DK-2300 Kbh. S
Denmark.

1. Introduction.

Eurotra is a research and development project in machine translation sponsored by the European Commission and the EEC member states. The project was launched in 1984, and its aim is to stimulate research in computational linguistics in Europe, and to produce a running prototype for a multi-lingual machine translation system towards 1990. This prototype will translate between any two of the nine official languages of the Communities within the subject field of information technology and have a dictionary of approximately 20.000 entries per language.

Most of what we shall say has been inspired by our work as Eurotra researchers, however, the views presented in this paper do not necessarily all reflect the official Eurotra position.

Acknowledgement.
We are indebted to those of our colleagues who have been investigating transfer problems for stimulating and thought-provoking papers on the topic.

2. The Translation System.

Eurotra is designed as a transfer based system. There are separate monolingual components for analysis and generation, and transfer components to link these. This means that we have monolingual components for Danish, Dutch, English, French, German, Greek, Italian, Portuguese, and Spanish and 72 transfer components to link these nine.

In analysis, the task of the monolingual component is to produce an abstract representation of the text. This we call an interface object because this representational object constitutes the input to the transfer component to either one of the other monolingual components. The target language generates a text on the basis of the output from the transfer component.

There are two important principles in the Eurotra design: compositionality and simple transfer. The translation process is compositional, i.e. the translation of a text is a function of the translation of its parts. Simple transfer basically means that the structure of the source language interface object is transferred unchanged to the target component, and that only lexical units change. Ideally, this should result in transfer components that only contain rules specifying the translation of source and target language lexical units, e.g.

know -> wissen
know -> kennen

for the translation of this English verb into German.

3. Disambiguation in bilingual MT-systems and in multi-lingual MT-systems.

In this paper, we shall present our ideas about how to develop a strategy for solving translation relevant lexical ambiguities in a multi-lingual machine translation system. Here it should be noted that in normal usage, a lexical unit is ambiguous if it has more than one denotation. In our usage, ambiguity is defined contrastively, that is a lexical unit is ambiguous if it has more than one translation into some other language. This was the case in the example already given for the translation of the English verb 'know' into either the German 'kennen' or 'wissen'.

For several reasons, an appropriate strategy for solving translation relevant lexical ambiguities in a multi-lingual machine translation system differs from that which may be adopted in a bilingual system. In a bilingual translation system, the semantic and syntactic similarities of and differences between the two languages can to some degree be accounted for by tuning the source and the target language grammars towards each other. Since the translation relevant ambiguities will be known, a high proportion of the disambiguation needed can be catered for in the source language component by entering a large number of specific readings for each lexical unit in the monolingual dictionary.

In a large multi-lingual system such as Eurotra where the same source language analysis result, i.e. the interface object, constitutes the input to eight different target languages, such a strategy has little attraction. Tuning the monolingual components towards each other would mean that the system would loose in extensibility not only with respect to extension of the grammars of the languages already part of the system, but also with respect to inclusion of new languages into it.

To sum up what has been said so far:

- Ambiguity is defined contrastively, in relation to another language.

- Analysis components should be developed monolingually and consequently such ambiguities cannot be taken into account.

- Transfer components should be kept as simple as possible.

That leaves the burden of disambiguation to the target language generation. As we shall see, this is not in conflict with the claim that generation components also should be developed monolingually. Actually, ambiguity arises bilingually, but can to a large extent be solved monolingually.

4. A strategy for disambiguation.

We propose a strategy where the basic principles are:

1) Disambiguation in analysis is restricted to disambiguation based on morphological criteria.

2) Disambiguation in transfer is restricted to those cases where we need access to information from the source language.

3) As the general principle, disambiguation is left to generation.

4.1. Disambiguation in analysis.

Disambiguation based on morphological criteria means that homographs belonging to different word classes, homograph nouns with different genders, and homographs from the same word class but with different inflection patterns are separated out into separate dictionary entries. This distinction automatically follows from the monolingual description necessary for morphological and syntactical analysis.

This means that we get 3 entries for 'like':

I **like** fish	- VERB
I never saw the **like**	- NOUN
People **like** you and me	- CONJUNCTION

Any other distinction is

- arbitrary
- not needed for monolingual description

4.2. Disambiguation in transfer.

Only in relatively few cases do we need access to information from the source language, and most cases can be handled just as well without access to such information.

One example where this information is needed is the translation of 'put' into German or Danish. The English verb is neutral as to horizontal or vertical position, whereas German and Danish have to make a choice between two verbs, 'stellen'/'stille' for vertical position, 'legen'/'lægge' for horizontal position. It is true that you also have the choice of a position-neutral verb like 'anbringen'/'anbringe' with a different stylistic value, corresponding to English 'place', but let us leave that out for the sake of the argument.

Now, if you have the German translations

'sie _____ die Flasche auf den Tisch'
'sie _____ das Buch auf den Tisch'

and you have to choose the right verb, you may in both sentences use 'stellen' as well as 'legen'. Only, bottles are <u>normally</u> placed in a vertical position on a table and books in a horizontal position, so if nothing was specified in the English text, you would choose the translations

'sie stellte die Flasche auf den Tisch'
'sie legte das Buch auf den Tisch'

Only if the English text had specified e.g. 'she laid the bottle on the table' or 'she stood the book on the table', would you choose the other possibilities, i.e.

'sie legte die Flasche auf den Tisch'
'sie stellte das Buch auf den Tisch'

Incidentally, this example is very dependent on the context. If the item is placed on a shelf, what is normal changes - books are normally put in a vertical position, whereas bottles are put in a horizontal position, at least in a wine cellar. So,

'she put the bottle on the shelf' (= 'on the rack')

translates into

'sie legte die Flasche in das Regal'

and

'she put the book on the shelf'

translates into

'sie stellte das Buch in das Regal'

If we could solve this ambiguity during generation, we would just need two simple rules for English -> German

put -> stellen
put -> legen

and correspondingly for English -> Danish

put -> stille
put -> lægge

and then leave it to generation to rule out the wrong translation. But we need the information that the source language had a neutral verb, and we also need information about the kind of object and about the place of location.

At present, we do not know how to distinguish between words like 'book' and words like 'bottle' nor how to distinguish between locations like 'on the table' and locations like 'on the shelf' in a systematic way. We shall probably have to write rather clumsy translation rules such as

put / _, obj | bottle... |, location | table... | -> stellen

put / _, obj | book... |, location | table... | -> legen

put / _, obj | bottle..|, location | shelf, rack..| -> legen

put / _, obj | book... |, location | shelf... | -> stellen

which should be read:
'put' translates into 'stellen', if 'put' is followed by an object which is a member of the set mentioned, and a location which contains a noun from the set mentioned.

These 4 rules should be regarded as exception rules to be tried first. If they do not apply, because the object is neither 'book' nor 'bottle', 2 simple rules will apply:

put -> stellen

put -> legen

and we shall get 2 translations of

'he put the newspaper on the table'

1 - 'er stellte die Zeitung auf den Tisch'

2 - 'er legte die Zeitung auf den Tisch'

Of these 2, the first one can be ruled out without having access to the source text, because newspapers not only normally are placed in a horizontal position, they always are - within our linguistic universe.

4.3. Disambiguation in generation.

As the general principle, disambiguation is left to generation. In one respect this is uneconomic because it means that we make more than one translation of ambiguous expressions, only to subsequently rule out the wrong one or the wrong ones. It would be more economic only to make the right translation, of course.

However, in another respect it <u>is</u> economic because in most cases a given ambiguity exists only in relation to some of the other 8 languages making up the system, and in these cases we can benefit from the similarity between the languages when there is no ambiguity.

If, for example, we want to translate the English verb 'adopt' into German, Danish and French, we have at least 3 translations into German and Danish:

1 - They **adopted** a child
- Sie **adoptierten** ein Kind
- De **adopterede** et barn

2 - He has **adopted** a new method
- Er hat eine neue Methode **eingeführt**
- Han har **indført** en ny metode

3 - The Council **adopted** the proposal
- Der Rat **verabschiedete** den Vorschlag
- Rådet **vedtog** forslaget

But into French we can use the same translation of the verb in all 3 cases:

1 - Ils ont adopté un enfant
2 - Il a adopté une nouvelle méthode
3 - Le Conseil a adopté la proposition

If we disambiguate in analysis, we get 3 entries in the English dictionary, adopt_1, adopt_2 and adopt_3. We then need 3 rules from English to French:

```
adopt_1 -> adopter
adopt_2 -> adopter
adopt_3 -> adopter
```

The French might have drawn the same distinction, and we would get:

```
adopt_1 -> adopter_1
adopt_2 -> adopter_2
adopt_3 -> adopter_3
```

If, however, we do not carry disambiguation this far in analysis, we can manage with only one rule from English to French:

```
adopt -> adopter
```

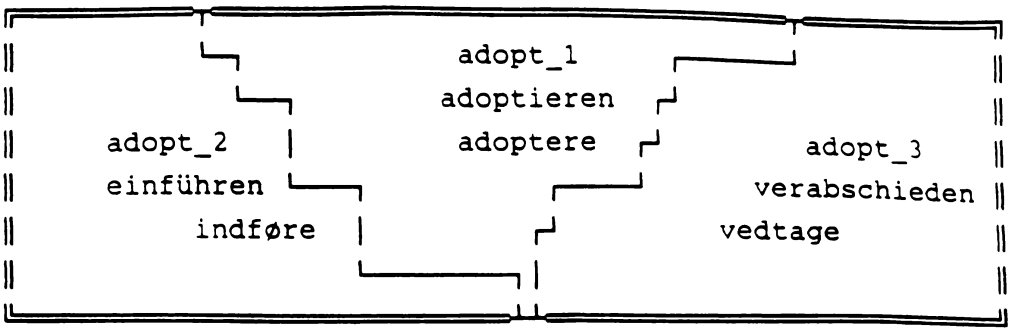
But would it not be convenient to have separate entries 'adopt_1', 'adopt_2' and 'adopt_3' for translating into German and Danish? -

```
          ┌ adoptieren
adopt_1   |
          └ adoptere

          ┌ einführen
adopt_2   |
          └ indføre

          ┌ verabschieden
adopt_3   |
          └ vedtage
```

This would work if 'adopt' always translates into 'einführen' and 'indføre', or into 'verabschieden' and 'vedtage', respectively, that is if the lexical structure of Danish and German were the same:



However, this is not the case, and furthermore the example is too simplified and more translations are needed than just three. In accordance with the principle of simple transfer, we prefer to leave the problem to generation and just write simple, context-free transfer rules:

```
English -> German : adopt -> adoptieren
                    adopt -> einführen
                    adopt -> verabschieden

English -> Danish : adopt -> adoptere
                    adopt -> indføre
                    adopt -> vedtage
```

So, we leave the problem to generation. Monolingually in the target language, we are presented with a choice of three different verbs:

```
    ┌ adoptierten ----┐
Sie ┼ führten --------┼ ein Kind  -- ein
    └ verabschiedeten ┘

                          ┌ adoptiert
Er hat eine neue Methode  ┼ eingeführt
                          └ verabgeschiedet

        ┌ adoptierte ----┐
Der Rat ┼ führte --------┼ den Vorschlag  -- ein
        └ verabschiedete ┘
```

and we must make a choice without having access to source language information.

In our example, the necessary rules may be formulated in the dictionary entries in the monolingual German dictionary:

(lu=adoptieren, sem_feat_object=+human,-adult)
(lu=einführen, sem_feat_object=+abstract v +concrete,-human)
(lu=verabschieden, sem_feat_object=+admin v +human,+adult)

(lu=Kind, sem_feat=+human,-adult)
(lu=Knabe, sem_feat=+human,-adult,+masculin)
(lu=Mädchen, sem_feat=+human,-adult,-masculin)
(lu=Methode, sem_feat=+abstract)
(lu=Vorschlag, sem_feat=+admin)
(lu=Beamter, sem_feat=+human,+adult)

'lu' is short for 'lexical unit'. This approach is based on a marking of all nouns with semantic features so that the selection of a verb can be made dependent on the semantic features of its arguments, i.e. its subject, direct object or indirect object.

The assignment of semantic features may create some problems. For instance, 'verabschieden' is not only a translation of the English verb 'adopt', but also of 'dismiss':

dismiss -> verabschieden

e.g.
'The manager dismissed the official'

translates into

'Der Direktor verabschiedete den Beamten'

However, this is also catered for by assigning two possible semantic feature sets of the object: either '+admin' or '+human,+adult'.

Developing a multi-lingual MT-system is a very delicate task. As has already been pointed out, it is important to have some very clear principles that are motivated and consistent, and that will hold not only for a small

prototype system but also allow for extension in terms of lexical coverage and in terms of inclusion of new languages. Yet at the same time, various pragmatic considerations are also necessary.

The implicit principle in the above discussion has been that disambiguation is carried out in generation and based on the semantic features of the context. However, what if we happen to have two linguistic expressions, two lexical units, following each other and both are ambiguous when translated into some language? Then the disambiguation of the first may depend on the semantic features of the second, and the disambiguation of the second may depend on the semantic features of the first. This might create an infinite loop.

Here we are helped by a compositional and context-free translation strategy, however. First all the parts of a sentence are translated, only then do we look at the various combinations. Sometimes this may create problems, but such problems are due to 'true' ambiguities, i.e. ambiguity in the normal usage of the term, that could not have been solved anyway. Suppose for example that we have the following rules from English into German:

```
discard -> verwerfen
discard -> verabschieden
master -> Lehrer
master -> Original (i.e. master copy)
```

and the following German dictionary entries:

```
(lu=verwerfen, semfeat_object= -animate)
(lu=verabschieden, semfeat_object=+human)
(lu=Lehrer, sem_feat=+human)
(lu=Original, sem_feat= -animate)
```

The English sentence:

She discarded the master

will in the first place, compositionally, get four translations:

Sie verwarf den Lehrer
Sie verwarf das Original
Sie verabschiedete den Lehrer
Sie verabschiedete das Original

Of these, two will be ruled out because there is no match between the semantic features of the verb and the object, and two will survive:

Sie verwarf das Original
Sie verabschiedete den Lehrer

The English sentence actually has these two meanings so we should get two translations. However, only one of these gives the intended meaning, but to find this, the system has to look beyond the sentence or to draw on information about text-type just as a human translator would. We shall not eleborate on that here.

In general though, we can rely on nouns being less ambiguous than verbs. This means that in practice we can to a large extent rely on the semantic features of nouns when disambiguating verbs. In the 'adopt' example above, there are no big problems in translating 'child', 'proposal', and 'method' into German, Danish, and French.

So far we have been concerned with contextually determined ambiguities. Within these, we may distinguish between

1. ambiguities that depend on the semantic context

and

2. ambiguities that depend on the syntactic context.

We have seen some examples of the first type and now we shall turn to the second.

The translation of the English verb 'know' into German, French, and Danish is dependent on whether its object is a clause or a noun phrase. Yet also here we can have context free translation rules:

```
English -> German:
     know -> kennen
     know -> wissen

English -> French:
     know -> connaitre
     know -> savoir

English -> Danish:
     know -> kende
     know -> vide
```

and monolingual dictionary entries:

```
German:
     (lu=kennen, object=np)
     (lu=wissen, object=clause)

French:
     (lu=connaitre, object=np)
     (lu=savoir, object=clause)

Danish:
     (lu=kende, object=np)
     (lu=vide, object=clause)
```

which will yield the correct translations:

```
                 ┌ Ich kenne die Frau
I know the woman ├ Je connais la femme
                 └ Jeg kender kvinden
```

I know that she is beautiful ┌ Ich weiss, dass sie schn ist
├ Je sais qu'elle est belle
└ Jeg ved, at hun er smuk

Here again, disambiguation in analysis would not really help us, as we would not get a one-to-one correspondence. 'know + clause' translates into 'savoir', but 'know + NP' may also translate into 'savoir', as in

I know my lesson -> Je sais ma lecon

Neither does the correspondence between 'savoir' and 'wissen'/'vide' hold here, as German and Danish use a modal verb 'knnen'/'kunne':

I know my lesson ┌ Ich kann meine Lektion
│
└ Jeg kan mine lektier

The translation of 'know' also demonstrates the need for a proper analysis of the text to be translated including resolution of pronoun references, as the criterion for the choice between 'kennen'/'wissen', 'connaitre'/'savoir' and 'kende'/'vide' is the structure of the antecedent of a pronoun, e.g.

She is beautiful, I know it ┌ Sie ist schn, das weiss ich
├ Elle est belle, je le sais
└ Hun er smuk, det ved jeg

She has a problem and I know it ┌ Sie hat ein Problem und ich kenne es
├ Elle a un probl
me et je le connais
└ Hun har et problem og jeg kender det

Apart from contextually determined ambiguities, we also have inherent ambiguities. This distinction should be seen as an operating distinction in an MT-system. It might be argued that there is an inherent semantic difference between 'adopt' in the sense 'adopt a child' and in 'adopt a proposal', but this is not really of much relevance so long as 'adopt' in the 'proposal'-sense can never take 'child' as an object, nor can 'adopt' in the 'child'-sense take 'proposal' as an object.

Operationally, we want to defer as much as possible to contextually determined ambiguities, as these are better controlled and more interesting from an MT point of view. What is left as inherent semantic ambiguities are consequently those cases where a word has more than one translation in the same context. Generally speaking, contextually determined ambiguities become inherent semantic ambiguities when the context is not informative enough. In these cases, disambiguation typically may be based on information about texttype.

E.g. the English noun 'pipe' translates into Danish 'fløjte', 'pibe' and 'rør'. In the following sentences, the context can be used for disambiguation:

She played the pipe -> Hun spillede på fløjte
She smoked a pipe -> Hun røg pibe
The pipe leaked -> Røret var utæt

However, a sentence like

8 pipes had been ordered

is translated into three equally correct sentences:

Der var blevet bestilt 8 fløjter
Der var blevet bestilt 8 piber
Der var blevet bestilt 8 rør

We must produce only one translation, and only one of the three translations actually convey the intended meaning. In cases like this we would have to apply a lexical preference mechanism, stating that in our text-type - information technology - the last translation is most likely to be the correct one. This mechanism might be based on the following text-type and dictionary information:

```
text-type=information technology > sem_feat=technology,...
text-type=arts  > sem_feat=literature, music, ...

(lu=fløjte, sem_feat=music)
(lu=rør, sem_feat=technology)
```

5. Final remarks.

To conclude, we sum up the principles of our strategy for solving lexical ambiguities in a multi-lingual machine translation system where we want to have the analysis and generation components developed monolingually and to keep the transfer components as simple as possible:

- Lexical disambiguation performed in the source language component is minimalistic in the sense that it is restricted to dealing only with morphologically based ambiguities, i.e. cases of homography where we can distinguish between separate lexical units on the basis of wordclass, gender, and/or inflectional pattern.

- Lexical disambiguation in transfer is restricted to those cases where the target language needs access to semantic information embedded in the source language lexical unit which is not recoverable to the target language on the basis of semantic and/or syntactic context.

- The rest of the disambiguation is to be resolved in target language generation.

From the point of view of efficiency, it might be claimed that a less restrictive approach to disambiguation in transfer would be preferable. Resolving more ambiguities in transfer means that as few translations as possible of a source language lexical unit are input to the target component, and the analysis and generation components can still be developed monolingually. However, such a strategy implies a vast increase in the size and the complexity of the transfer components - the number of which will always be much greater than that of monolingual components in a multi-lingual system. Therefore, having the target language disambiguate according to the strategy we have outlined here appears to us to be the soundest approach. As we have argued and exemplified, a large number of different types of lexical ambiguity problems lends themselves to being resolved in the course of target language generation in accordance with the principle of truly monolingually based language components.

ATT KNYTA NORDENS SPRÅK TILL ETT MÅNGSPRÅKIGT DATORÖVERSÄTTNINGSSYSTEM

Klaus Schubert

BSO/Research
Postbus 8348
NL-3503 RH Utrecht
Nederländerna

Uppsala universitet, FUMS
Box 1834
S-751 48 Uppsala
Sverige

Elektronisk adress: schubert@dltl.uucp

1. Ett mångspråkigt datoröversättningssystem

Jag beskriver här i korthet datoröversättningssystemet *Distributed Language Translation* (**DLT**) och tar i samband med detta upp Nordens språk. DLT är ett omfattande forsknings- och utvecklingsprojekt som bedrivs av det nederländska softwareföretaget Buro voor Systeemontwikkeling (BSO/Research i Utrecht) med anslag från Nederländernas Ekonomidepartement. Projektet är inne på en än så länge icke-kommersiell sjuårsperiod (1985-1991) som skall leda till en prototyp för ett översättningssystem för icke-litterär engelska och franska. Prototypen omfattar bara två språk, men DLT är från början beräknat att bli **mångspråkigt**, vilket innebär att det måste vara **modulärt utbyggbart**. Därför utförs redan nu i samarbete med forskare vid universitet i vederbörande länder och med andra experter förberedande studier om tillämpligheten av DLT:s grammatikmodell på andra språk (och även implementeringar i begränsat omfång). Bland dessa är även några av Nordens språk.

Det är möjligt att utvecklingen av utgångs- eller målspråkssystem för fler språk påbörjas inom DLT före 1991. Med detta perspektiv upptar jag här frågan om hur Nordens språk kan knytas till DLT.

2. Utbyggbarhet och spridningen

DLT:s översättningsmetod är styrd av två förutsättningar: utbyggbarhetskravet och den idé från vilken beteckningen *Distributed* härrör: spridningen i översättningsprocessen.

Utbyggbarhetskravet gör det nödvändigt att skapa ett väldefinierat interface till vilket godtyckliga utgångs- och målspråk kan knytas, utan att redan befintliga delar av systemet för den skull behöver anpassas. I DLT är detta interface ett **mellanspråk**. DLT:s mellanspråk är en något modifierad version av esperanto. Valet av esperanto

har motiverats i andra arbeten (Witkam 1983; Schubert 1986b, Schubert u.u. b). Orsakerna kan sammanfattas mycket skissartat i en jämförelse med de tre andra typer av teckensystem som skulle kunna tänkas fungera som mellanspråk:

Esperanto lämpar sig för mellanspråksfunktionen i ett datoröversättningssystem bättre än

1. **folkspråk**, eftersom mellanspråket måste vara syntaktiskt oambiguöst, och folkspråken är på språktecknets formsida för oregelbundna;
2. **formella symbolsystem**: Eftersom mellanspråket som enda förbindelselänk mellan utgångs- och målspråk måste återge textens fullständiga innehåll med alla nyanser, är konstgjorda system genom själva sin beskaffenhet otillräckliga (jfr Hjelmslev 1963: 101);
3. **andra planspråk** (volapük, ido, novial, interlingua m fl) eftersom mellanspråket måste äga ett **autonomt** semantiskt system som är oberoende av utgångs- och målspråken. Ett sådant system, som gör ett konstgjort och i början referensspråksberoende system till ett självständigt mänskligt språk, kan inte skapas, utan det kan bara uppstå genom långvarigt oreflekterat bruk av språket i en tillräckligt stor språkgemenskap. Av alla planspråksprojekt har bara esperanto genomgått denna utveckling fullständigt (Blanke 1985: 107ff, särskilt 112 tabell 2; jfr även Bagger 1986: 16ff).

Den andra förutsättningen som karaktäriserar DLT är spridningen. Enligt planerna skall DLT fungera i datakommunikationsnät. Man skall därför inte föreställa sig DLT som en översättningsmaskin som står i ett rum och producerar snyggt tryckta översättningar av texter som matas in i den. DLT kan bäst ses som en softwarekomponent inom kontorsautomatiseringen som gör befintliga servicetjänster mångspråkiga. Till exempel on-line databankanlitning.

Det är allmänt vedertaget att **hel**automatisk översättning av hög kvalitet är omöjlig. Översättningsdatorn måste alltid få hjälp av en människa. Detta gäller också för DLT. I och med att översättning med mellanspråk på sätt och vis är dubbel översättning, så skulle man egentligen behöva en medhjälpare för utgångsspråket och en för varje målspråk. Men spridningstanken förbjuder detta i praktiken. När man tänker på tillämpningar som databankanlitning och liknande, så är det för dyrt att genast översätta alla texter som matas in i databanken till alla systemets målspråk. Man skulle då vara tvungen att lagra varenda text i så många kopior som man har målspråk, och man skulle dessutom vara tvungen att översätta hela databanken varje gång man lägger till ett nytt målspråk. Därför är det mera praktiskt att översätta varje text med en enda människas hjälp bara till den grad att den kan översättas vidare helautomatiskt så snart en kund vill läsa en bestämd text på ett bestämt målspråk. Hela databanken byggs upp av texter i denna halvöversatta form. Halvvägsprodukten är just en text på mellanspråket. Samma text kan i ett sådant system behöva översättas flera gånger till samma målspråk om olika kunder vid olika tider beställer den. I en sådan spridd uppställning kan man inte ha en medhjälpare för varje målspråksmodul. Därför behövs ett mycket speciellt mellanspråk som möjliggör helautomatisk vidareöversättning. Detta är bland annat vad de tre kriterierna ovan beskriver.

3. Språkspecifikt och språkövergripande

Mellanspråket har fler funktioner än bara att vara en halvvägsprodukt, och dessa funktioner är avgörande för de villkor under vilka man kan knyta flera språk till systemet.

En text på ett godtyckligt språk kan översättas till vilket annat språk som helst. Betraktar man utgångstexten som bestående av språktecken med form och innehåll, så innebär denna tes att det till utgångsspråkets former finns motsvarande former i målspråket med (mer eller mindre) samma innehåll. Detta gäller visserligen inte mellan enstaka ord i två språk, men ganska väl mellan hela texter på de två språken. På så sätt kan man säga att formen är språkspecifik, medan innehållet är språköver-gripande. Översättningsprocessen skall förändra formen, men bibehålla innehållet.

Detta är en mycket förenklad framställning. I själva verket är innehållet tyvärr **uppdelat** på olika sätt i olika språks semantiska system (Schubert 1987: 200), så att inte heller innehållet utan vidare kan anses vara den språkoberoende nivå på vilken transfersteget i översättningen kan tas. Det är detta som gör automatisk översättning så svår. Man bör emellertid noggrant fastslå vad svårigheten består i: Problemet är inte att språkoberoende innehåll inte finns, utan att man inte kan skriva upp och hantera rent innehåll på ett språkoberoende sätt. Detta är problemet med betydelse-representationen. Vilket teckensystem man än hittar på för detta ändamål, är det antingen ett mänskligt språk eller också är det beroende av ett sådant. Detta följer av Hjelmslevs (1963: 101) översättbarhetskriterium.

Konsekvensen har för DLT varit att välja själva mellanspråket som betydelse-representation (Schubert 1986c: 146ff). Efter hundra års utveckling, är esperanto på innehållssidan ett fullgott mänskligt språk, men det har kvar den formella regelbundenhet som är typisk för ett planspråk.

En annan konsekvens är ännu väsentligare för anknytningen av nya språk till systemet: Eftersom DLT:s betydelserepresentation är mellanspråket, har all semantisk och pragmatisk bearbetning förlagts till mellanspråket. Jag beskriver nedan hur detta går till. Men det är på plats att redan här nämna vad det intressanta med detta tillvägagångssätt är. Lyckas man förskjuta alla processer som har med betydelse att göra till mellanspråket, så behöver man bara utveckla system för dessa tunga och komplicerade processer en enda gång. De kan sedan fungera i systemet för översättningar från och till vilka språk som helst.

Om denna idé är genomförbar så är man framme vid ett ganska smalt och väldefinierat interface till godtyckliga utgångs- och målspråk: En **kontrastiv syntax** och ett **tvåspråkigt lexikon**, båda med mellanspråket som ett av språken.

I stora drag kan översättningsförloppet då beskrivas så här: Först en språkspecifik syntaktisk analys i utgångsspråket. Förekommande alternativ där valet bara kan träffas med hjälp av semantik och pragmatik tas med parallellt till mellanspråket. I systemets kärna utförs de språkövergripande semantiska och pragmatiska bearbetningarna. Till slut följer språkspecifika syntessteg i målspråket. Detta beskrivs mera i detalj i nästa avsnitt.

4. Metatax

Det är omöjligt att beskriva hela översättningsprocessen i ett datoröversättningssystem på ett par sidor. För DLT ges sådana beskrivningar vid olika utvecklingsstadier av systemet av Witkam (1983: III-46ff), Papegaaij (1986: 75ff) och mig (Schubert 1986c: 126ff). Sedan dessa arbeten kom till har DLT förändrats på viktiga punkter. Det nuvarande läget är mera stabilt, eftersom systemet DLT sedan oktober 1987 finns inte bara i lösa moduler, utan också i en sammankopplad preliminär prototyp för översättning från engelska över esperanto till franska, något som i Utrecht i december 1987 uppvisades för fackpressen. Jag skall här försöka ge en snabb genomgång av översättningsprocessen och ta upp särskilt de delar som spelar en roll för anknytningen av nya utgångs- och målspråk.

4.1. Dependenssyntaktisk parsning

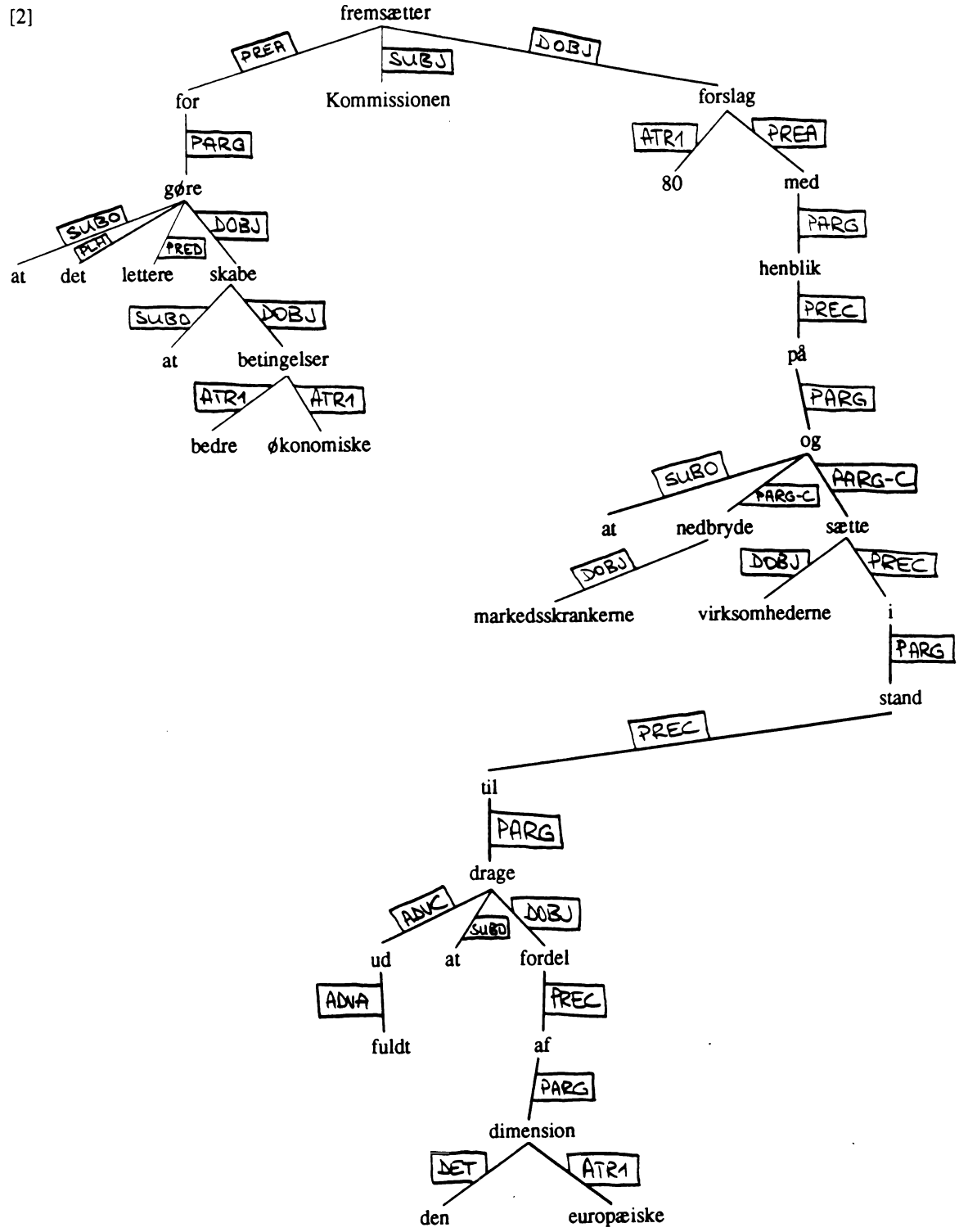
Första steget i DLT-översättningen är en parser som utför den syntaktiska analysen av den inmatade utgångstexten. Inom DLT används en mycket bred definition av begreppet *syntax* som omfattar språktecknets hela formsida. Syntax är alltså formernas grammatik både på ord-, menings- och textnivå. Resultatet av den syntaktiska analysen återges i form av **dependensträd**. Vi har för DLT valt dependenssyntaxen och anpassat den till datalingvistiska behov (se om dependensgrammatiken Tesnière 1959/1982; Nikula 1986; Schubert u.u. a; om DLT:s modell Schubert 1986a: 14ff, 1987: 28ff). Dependensträd bygger inte upp någon abstrakt struktur ovanpå orden, utan har själva orden på noderna. Över varje ord finns en etikett som anger dess syntaktiska funktion i förhållande till det styrande ordet. Ett sådant träd för mening [1] ser ut som [2] på nästa sida.

[1] For at gøre det lettere at skabe bedre økonomiske betingelser, fremsætter Kommissionen 80 forslag med henblik på at nedbryde markedsskrankerne og sætte virksomhederne i stand til fuldt ud at drage fordel af den europæiske dimension.

Det finns flera orsaker till att föredra dependenssyntax framför konstituensmodeller (Schubert 1987: 193f). Den viktigaste anledningen går ut på att dependensanalysen mera direkt kommer fram till just de drag i textens struktur som är väsentliga för översättningen. Dependenssyntaxen behandlar den syntaktiska **funktionen** primärt och den syntaktiska **formen** sekundärt. I konstituenssyntaxen är det tvärtom.

För att sedan kunna finna orden i ett tvåspråkigt lexikon måste de kunna brytas ned till grundformerna: nominativer, singularisformer, infinitiver osv. Detta kan göras antingen i parsern eller senare med hjälp av redundansregler för lexikonet. I båda fallen måste man anteckna vilka drag ordet hade innan det bröts ned. Men vilka drag måste tas med? Jag diskuterar ovan att man försöker att åtskilja språkspecifika och språkövergripande egenskaper hos ord och ordgrupper och jag talar i det sammanhanget mycket om språktecknet. Det är nyttigt att tänka på tecknet även när det gäller syntaktisk parsning (Schubert 1987: 152ff). Alla syntaktiska (med morfologiska, ordbildningsmässiga m fl) drag är nämligen inte språktecken, dvs de har inte form **och** innehåll. Substantivets numerus har teckenfunktion. Står ett ord i pluralis så översätts det vanligtvis också med ett pluralisord. Men till exempel kasus är

[2]


fremsætter
PREA
SUBJ
DOBJ
for
Kommissionen
forslag
PARG
ATR1
PREA
gøre
80
med
SUBO
DOBJ
PARG
PLH
PRED
at
det
lettere
skabe
henblik
SUBO
DOBJ
PREC
at
betingelser
på
ATR1
ATR1
PARG
bedre
økonomiske
og
SUBO
PARG-C
PARG-C
at
nedbryde
sætte
DOBJ
DOBJ
PREC
markedsskrankerne
virksomhederne
i
PARG
stand
PREC
til
PARG
drage
ADVC
SUBO
DOBJ
ud
at
fordel
ADVA
PREC
fuldt
af
PARG
dimension
DET
ATR1
den
europæiske

inte något språktecken. Det är form utan något direkt översättbart innehåll. Kasus pekar på en viss syntaktisk funktion, och den kan i sin tur sedan översättas. Man kan inte gärna beskriva hur en tysk genitiv översätts till svenska utan att ta omvägen över den syntaktiska funktionen. Är genitivordet attribut (*die Last der Schulden*), objekt (*sich der Schulden entledigen*) eller prepositionsargument (*trotz der Schulden*)?

Den syntaktiska analysen levererar ett träd med orden, deras syntaktiska funktioner och översättningsrelevanta drag till nästa steg i processen: den kontrastiva syntaxen. Finns det i utgångsmeningen ambiguiteter som inte kan lösas med enbart syntaktiska medel, så genereras alla syntaktiskt möjliga lösningar parallellt och överförs parallellt till nästa steg.

4.2. Översättningssyntax

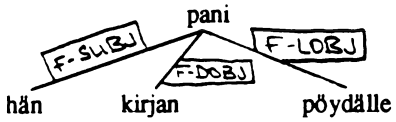
Det andra stora bearbetningssteget i DLT-processen är **metataxen**. Ordet är taget från Lucien Tesnières (1959/1982: 283) term *métataxe* för den syntaktiska förändring som utförs i en text eller mening under översättningsprocessen. Metataxen är på samma sätt som parsningen helt formorienterad. Metataxmodulen producerar **alla syntaktiskt möjliga översättningar** till mellanspråket av alla alternativa träd som kom ut ur parsern.

Metataxprocessen kan göras rekursivt, så att antalet kontrastiva syntaktiska regler inte blir oändligt. Formellt sett är hela metataxen trädmanipulation. Man kan bäst bearbeta komplexa träd för hela meningar med bisatser och liknande genom att omforma trädet bit för bit uppifrån och ned. Metataxregler består av ett utgångsspråksträd och ett mellanspråksträd. Trädet för den utgångsspråksmening man vill översätta jämförs med metataxreglernas utgångsspråksträd. Man börjar vid trädets högsta nod (vanligtvis det finita verbet) och söker en metataxregel som i sitt utgångsmönster har den noden, eller en variabel som kan stå för den noden. Har man funnit en sådan regel, så ersätter man den biten av det träd man håller på att omforma med det mellanspråksmönster som ges i regeln. På så sätt uppstår **hybrida träd** som har en **gräns** mellan ord och syntaktiska etiketter från båda språken. Har man utfört en sådan ersättningsoperation, söker man nästa ännu oomformade symbol (ord eller etikett) under gränsen och försöker finna en passande metataxregel.

Här är en finsk mening med sitt träd. [5] är en esperantomotsvarighet till [3]. Metataxen skall omforma [4] till [6]. [7] är ett av de hybrida träd som uppstår halvvägs under metataxprocessen. Givetvis har mening [3] fler översättningsalternativ, bl a ett med betydelsen *li* 'han' för det finska *hän*. Det är en intressant textgrammatisk uppgift att lösa denna pronominala ambiguitet.

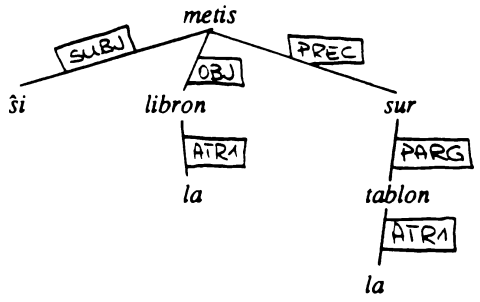
[3] Hän pani kirjan pöydälle.
'han/hon lade boken på-bordet'

[4]

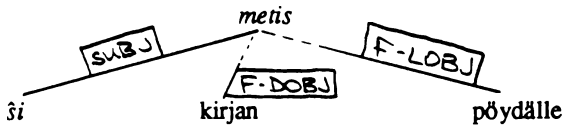


[5] *Ŝi metis la libron sur la tablon.*

[6]



[7]



Eftersom metataxen inte är någon urvalsprocess utan skall producera alla syntaktiskt möjliga översättningsalternativ, finns det omarkerade metataxregler som ser till att det alltid finns en regel som passar. Metataxprocessen får aldrig förkasta en mening eller ett översättningsalternativ. Omarkerade regler kan dock inte helt och hållet lösa problemet. Givetvis måste det också finnas ett tvåspråkigt lexikon. Men om ordet finns i lexikonet, så måste det alltid finnas en regel som omformar en syntaktiskt korrekt struktur som innehåller detta ord.

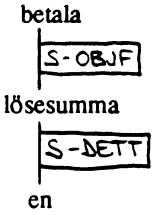
Ett metataxinriktat lexikon har i princip samma struktur som metataxreglerna: Ett träddiagram i utgångsspråket som omformas till ett mellanspråksträd. I många fall består sådana "träd" i lexikonet bara av en enda nod med ett enda ord på, men som i vanliga ordböcker så är det ofta nödvändigt att översätta från eller till en ordgrupp som enhet. Dessa har då en mera egentlig trädform (exemplen på svenska och esperanto är tagna från Munniksma m fl 1975):

[8] indexlån -> *indeksita prunto*

[9]

[10] betala en lösesumma -> *liberaĉeti*

[11]



Det som står i lexikonet är transformationsregler för delträd, som utför precis samma

sorts process som metataxreglerna. I princip finns faktiskt ingen skillnad mellan transformationsreglerna i lexikonet och i metataxregelsystemet. Det är fråga om en någorlunda godtycklig gränsdragning mellan de två. Detta beror på att metataxreglerna är **redundansregler** i (eller före) lexikonet. Man skulle teoretiskt kunna ha all transformationsinformation i lexikonet, vilket skulle innebära en ofantlig mängd upprepningar på grund av språkets regelbundenhet. Man har för effektivitetens skull tagit ut ur lexikonet allt som kan formuleras mera generellt i allmännare regler. Dessa är de egentliga metataxreglerna.

Trots att det kan låta enkelt, är det ganska invecklat att beskriva alla tänkbara syntaktiska omformningar mellan två språk i pålitliga regler. Regelsystemet kan därför bli relativt stort och det är viktigt att inskränka reglernas antal så mycket som möjligt, men att ändå hitta rätt regel snabbt.

För att begränsa regelantalet finns det en effektiv metod: Även om parsern gallrar bort sådana syntaktiska drag som bara identifierar syntaktiska funktioner och som därför inte spelar någon roll i själva översättningen så snart dessa funktioner har gjorts explicita i etiketter på trädets grenar, så kan det ändå finnas alternativa strukturer som har samma översättning. Till exempel stavningsalternativen da. *i dag* (två noder) och *idag* (en nod). För att slippa skriva metataxregler för alla sådana alternativ kan man i metataxreglerna införa **utgångsspråksfilter**. Dessa är metataxregler som omformar utgångsspråkliga träd till andra träd i samma språk. Det är alltså ett slags förberedande bearbetning. I exemplet skulle man då omforma *i dag* till *idag* (eller omvänt) och ha en översättningsregel till esperanto bara för den ena formen. Är det då så att sådana formalternativ verkligen betyder exakt samma sak? Finns det inte någon stilskillnad som man vill ha med i översättningen? Metataxmetodiken tillåter att i detta sammanhang återigen behandla sådana skillnader som indikatorer för något annat, men inte som skilda språktecken. Detta innebär att två alternativa former visserligen översätts på samma sätt, men att skillnaden ändåkan ha en funktion, exempelvis som stilmarkör (jfr sv. *ska/skall*, eng. *don't/do not*). Utgångsspråksfiltren kan anteckna detta och ge stilinformationen till vederbörande textgrammatiska stilregel, och ändå översätta båda formerna likadant på meningsnivå.

Utgångsspråksfiltren gör det också möjligt att förbereda det träd som skall översättas. Man skiljer ju vanligtvis på innehållsord och funktionsord. Somliga funktionsord översätts normalt inte direkt med ord, utan med syntaktiska drag på vissa innehållsord, till exempel hjälpverb. Man kan därför skriva utgångsspråksfilter även för detta. I stället för två noder för *har kommit* får man då en nod med *komma* och draget "[perfekt]". Hur långt man skall gå med dessa förberedande omformningar beror i princip på ett godtycklig beslut, men godtyckliga beslut, som är mycket vanliga i grammatiskt systembyggande, skall givetvis vara ändamålsenliga. En "naturlig" gräns till förberedningsomformningarna är lexikonet: För de egentliga transformations-reglerna (alltså för de regler som inte filtrerar inom samma språk utan ersätter träddelar med ord och etiketter från det andra språket) är det viktigt att trädet som översätts och träden i lexikonet förblir jämförbara. Man får alltså genom förberedande filter bygga om trädet bara så mycket att det ännu är jämförbart med de träd som en lexikograf finner naturligt att använda (Schubert 1987: 178).

Jag har redan antytt hur transformationsreglerna fungerar: De bildar hybrida träd

genom att uppifrån och ned söka utgångsspråkliga element i trädet och ta fram passande metataxregler för att omforma dessa element till mellanspråket. Dessa transformationsregler kan givetvis inte alltid behandla ett isolerat ord eller en isolerad syntaktisk etikett. Oftast beror valet av en bestämd transformationsregel på ordets omgivning. Hur väljs den rätta regeln ut? Metoden är ganska enkel: Man anger omgivningen i metataxregelns utgångsmönster. Vill man säga att ett svenskt objekt vanligtvis översätts till esperanto som objekt, så behöver man bara en regel som uttrycker detta:

[12]

Men vill man uttrycka att objektet till passiva verb översätts enligt en annan regel (*han* [subjekt] *serveras kaffe* [objekt] -> *al li servatas kafo* [subjekt], ordagrant: 'till honom serveras kaffe'), så skall dessa omgivningsvillkor uttryckas i regeln:

[13]

Så som reglerna [12] och [13] är formulerade här passar de båda två på passivmeningen *han serveras kaffe*. Ändå används bara [13] utan att detta behöver anges i reglerna. Detta är en praktisk åtgärd som tillåter att formulera allmänna regler som [12], utan att man behöver förändra dem varje gång man skriver en mera speciell regel för något särfall. För att detta skall fungera inför man en **hierarki** på metataxreglerna: Den regel har förtur vars utgångsmönster är mera specifikt. Detta mäts på två sätt: Ett bokstavligen utsatt ord är mera specifikt än en variabel för en hel ordklass, och en struktur med fler noder och etiketter är mera specifik en en sådan med färre element. Det andra villkoret ger regel [13] förtur så att [12] inte används. Det första villkoret ger helt allmänt lexikonreglerna (som har ett ord som högsta nod i sitt utgångsmönster) förtur framför de egentliga metataxreglerna (som har en variabel som högsta nod), så att undantag kan tas om hand i lexikonet. Ibland vill man få fram parallella, strukturellt olika översättningsaltvernativ. I sådana fall kan man ge en metataxregel statusen "parallell", vilket innebär att också nästa regel i förturshierarkin används.

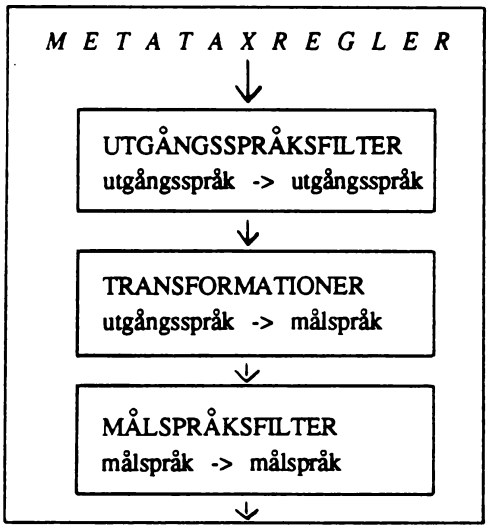
Hierarkin är något man inför i regelsystemet för att hålla det överskådligt och för att kunna bygga ut det på ett hyggligt sätt utan att varje förändring har svårförutsedda biverkningar. En likadan säkerhetsåtgärd är filtreringen **efter** transformationsreglerna. Regelsystemet är trots alla strukturering så komplext som en kontrastiv syntax brukar vara. Därför är det praktiskt om man kan tillåta vissa övergenerella transformationsregler. Regel [12] till exempel genererar ett objekt under vilket esperantoverb som helst. Men det är ju möjligt att lexikonet någon gång översätter ett transitivt svenskt

verb med ett intransitivt verb på esperanto. För att ändå undvika fel i sådana fall kan man antingen förse varje transformationsregel med ett speciellt korrekthetstest, eller också kan man filtrera bort felen efteråt. Den senare lösningen har valts i DLT:s metataximplementering: Mellanspråksfilter kontrollerar det genom transformationerna genererade trädets korrekthet. Till detta behövs två sorts filter: Det ena kontrollerar om en viss etikett får befinna sig under ett visst ord. Frågan är alltså t ex: Kan det här verbet styra ett objekt? Den andra sortens filter kontrollerar om orden får stå under sina etiketter. T ex: Får en infinitiv vara subjekt? Eftersom metatax inte gör något urval, får underkända konstruktioner inte förkastas utan måste omformas till korrekta träd. Vid detta tillfälle läggs vissa syntaktiska drag till orden, som följer ur de syntaktiska funktionerna. Objekten får sina ackusativändelser osv.

När den egentliga metataxen är färdig så har man ett träd (eller flera alternativa träd) vari alla ord är omsatta till mellanspråket, alla ord står på rätt plats i trädet och vissa syntaktiska drag är utsatta. Nu följer en annan sorts efterfiltrering: rektions- och kongruensregler. Dessa regler sprider redundanta drag över trädet. Ett objektsubstantivs attributiva adjektiv får nu också sina ackusativändelser osv. Nu föreligger en korrekt mellanspråksmening i trädform. Senare kommer de syntaktiska etiketterna att tas bort och trädet omformas till en mening. Efter denna trädlinearisering får man en vanlig text på mellanspråket.

Metataxsystemet är uppbyggt på följande sätt:

[14]



4.3. Semantik och pragmatik

Innan det sista steget, trädlineariseringen, kan tas måste en enda rätt översättning väljas ut ur mångfalden av syntaktiskt möjliga alternativ, generade av metataxen. Detta sker i DLT genom ett **semantisk-pragmatiskt ordexpertsystem** som bygger på en **kunskapsbank**. Båda delarna är skrivna enbart på mellanspråket esperanto. Eftersom metataxmetodiken innebär att alla semantiska och pragmatiska bearbetningar går ut på

att välja bland redan befintliga alternativa esperantomeningar, så kan samma expertsystem med samma kunskapsbank utföra bearbetningarna oberoende av vad utgångsspråket var. Detta är givetvis bara möjligt om mellanspråket är autonomt och inte tillyxat efter vissa utgångsspråks modeller (jfr avsnitt 2). Jag tar här inte upp detaljer om ordexpertsystemet. Det har beskrivits mycket utförligt av Papegaaij (1986: 75ff).

Efter det semantisk-pragmatiska urvalet kan tveksamma alternativ kvarstå. Kan systemet inte med tillräcklig säkerhet komma fram till en enda översättning, så rådfrågas människan i en **interaktiv dialog**. Här parafraseras de kvarstående alternativen på utgångsspråket, så att den som arbetar med DLT varken behöver vara översättare eller behöver behärska vare sig esperanto, målspråken eller dependensgrammatiken. DLT är ett användarsystem, inte någon översättarhjälp.

4.4. Från mellanspråk till målspråk

Har ordexpertsystemet, eventuellt med människans hjälp, valt ut den riktiga översättningen, kan trädet linearisera och den första hälften av översättnings-processen är färdig. Texten kan nu läsas på esperanto. Den kodas för nätöverföringen och mottas i denna form av en mottagarmodul. Där avkodas den och analyseras i en parser, vilket går snabbt eftersom DLT:s mellanspråk är syntaktiskt oambiguöst. Härefter följer metataxen, den här gången från mellanspråket till målspråket.

Den enda väsentliga skillnaden mot första hälften av processen ligger i den semantiska bearbetningen. Här används samma ordexpertsystem för att välja ut den bästa översättningen i målspråket. Hur kan det vara möjligt när ordexpertsystemet är gjort för esperanto och inte för målspråket? Den lösning som möjliggör detta bygger på en idé som är mycket vanlig i normala ordböcker för mänskligt bruk: Har ett utgångsspråksord flera översättningar så anges i ordböcker ofta i vilken kontext vilken av översättningarna skall väljas. Denna kontextuppgift ges normalt på utgångsspråket. På samma sätt fungerar de tvåspråkiga lexikonen för målspråk inom DLT.

Detta är ett exempel på en ingång i DLT:s esperanto-franska lexikon. Först står utgångsordet (esperanto, med morfemgränser angivna), sedan en semantisk relator (ett esperantomorfem) och sedan en rad esperantoord som anger en typisk omgivning i vilken det franska målordet skall användas som ges i sista spalt. Därunder ger jag ungefärliga svenska glosor (*akr'a* betyder bl a 'skarp').

[15]

'akr'a', ['a', *'dolor'o, mal'varm'o, riproĉ'o'j, vort'o'j, romp'o, eĝ'o'*], [**'vif'**/'ADJ']
smärta, köld, förebråelser, ord, brytning, kant
'akr'a', ['a', *'naz'o, orel'o'j, tur'o'*], [**'pointu'**/'ADJ']
näsa, öron, torn
'akr'a', ['a', *'spic'o, pipr'o, brand'o'*], [**'fort'**/'ADJ']
krydda, pepper, brännvin
'akr'a', ['a', *'disput'o, batal'o, kriz'o, vent'o, ŝtorm'o'*], [**'violent'**/'ADJ']
dispyt, strid, kris, vind, storm
'akr'a', ['a', *'ironi'o'*], [**'mordant'**/'ADJ']
ironi

...

På samma sätt som ordexpertsystemet i översättningsprocessens första hälft väljer ut

det i kontexten mest sannolika alternativet, så gör det det också nu, nämligen genom att jämföra olika alternativs kontextförväntningar med den aktuella kontexten i den text som översätts. Dessa jämförelser utförs i esperantotexten som föreligger fullständig och i en av människan i dialogen disambiguerad och godkänd form.

Har den rätta översättningen på så sätt valts ut, så kan den lineariseras till en vanlig text och översättningen är fullbordad.

5. Hur knyter man fler språk till systemet?

Den skissartade genomgången av översättningsförloppet i systemet DLT visar i grova drag DLT:s modulära struktur. Vad måste nu till för att knyta ett nytt utgångs- eller målspråk till detta system?

Det är ett enda regelsystem som knyter ett språk till DLT:s kärna: metataxen. Detta system omfattar de egentliga metataxreglerna och ett tvåspråkigt lexikon med mellanspråket. En metatax förutsätter en dependenssyntax för språket i fråga (och för mellanspråket). Metataxen med lexikonet behövs vare sig det nya språket skall vara utgångs- eller målspråk, men lexikonen är något olika i de två fallen. För att kunna tillämpa ett metataxsystem på ett utgångsspråk måste det dessutom finnas en parser som analyserar den inmatade texten enligt samma dependenssyntax som metataxreglerna bygger på, och en dialogmodul. För ett nytt målspråk krävs givetvis ingen parser, men däremot trädlineariseringsregler. I båda fallen behövs för ett nytt språk först en dependenssyntax med ett syntaktiskt lexikon och sedan en metatax med ett tvåspråkigt lexikon.

De ovan nämnda förstudierna för anknytningen av fler språk till DLT gäller då i första hand dependenssyntaxer och sedan metataxer. Vad gäller Nordens språk så har dependenssyntaxer enligt DLT-modell utarbetats (med ännu inte publicerats) för finska av Kalevi Tarvainen (1987) och för danska av Ingrid Schubert (1987). En isländsk arbetsgrupp har just bildats. Intressenter för övriga språk är välkomna. Från språkområden utanför Norden har DLT redan fått en tysk och en ungersk dependenssyntax (Lobin 1987; Prószéky/Koutny/Wacha 1987), medan arbetsgrupper på olika håll har börjat arbeta med en rad romanska, slaviska och östasiatiska språk. Flera väntas tillkomma under 1988.

Noter

1. Träddiagrammen i denna uppsats är tänkta som illustrationer. Givetvis kan jag här inte förklara i detalj de bakomliggande syntaxbeskrivningarna. Jag ber läsaren om ursäkt för att jag citerat opublicerat material om finskan och danskan. Angående finskan har jag också lånat en exempelmening från Tarvainen (1985: 166). De svenska träden är mina egna preliminära förslag, och esperantoträden motsvarar DLT:s mellanspråkssyntax (Schubert 1986: 23ff).

2. Jag tackar Lena Odén för språkgranskningen.

LITTERATUR

Bagger, Preben (1986): *Sprog og sprog imellem.*
Skelby: Kommunikation og Kultur

Blanke, Detlev (1985): *Internationale Plansprachen.*
Berlin: Akademie-Verlag

Hjelmslev, Louis (1963): *Sproget.*
København: Berlingske forlag (2:a uppl.)

Lobin, Henning (1987): Dependenzsyntax des Deutschen.
Opublicerad rapport. Utrecht: BSO/Research

Munniksma, F. m fl (1975): *International business dictionary in nine languages // Internacia komerca-ekonomika vortaro en naŭ lingvoj.*
Deventer / Antwerp: Kluwer

Nikula, Henrik (1986): *Dependensgrammatik.*
Malmö: Liber

Papegaaij, B. C. (1986): *Word expert semantics. An interlingual knowledge-based approach.*
Utg. V. Sadler / A. P. M. Witkam. Dordrecht / Riverton: Foris

Prószéky, Gábor / Ilona Koutny / Balázs Wacha (1987): A dependency syntax of Hungarian (for use in DLT).
Opublicerad rapport. Utrecht: BSO/Research

Schubert, Ingrid (1987): Dänische Dependenzsyntax für DLT.
Opublicerad rapport. Utrecht: BSO/Research

Schubert, Klaus (1986a); *Syntactic tree structures in DLT.*
Utrecht: BSO/Research

Schubert, Klaus (1986b): Wo die Syntax im Wörterbuch steht. Esperanto als Brückensprache der maschinellen Übersetzung.
i: *Pragmantax.* Utg. Armin Burkhardt / Karl-Hermann Körner. Tübingen: Niemeyer, s. 449-458

Schubert, Klaus (1986c): Linguistic and extra-linguistic knowledge.
i: *Computers and Translation* 1, s. 125-152

Schubert, Klaus (1987): *Metataxis. Contrastive dependency syntax for machine translation.*
Dordrecht / Providence: Foris

Schubert, Klaus (under utgivning a): Inbjudan till en tillämplig språkteori.
i: *Nysvenska studier*

Schubert, Klaus (under utgivning b): Ausdruckskraft und Regelmäβigkeit. Was Esperanto für automatische Übersetzung geeignet macht.
i: *Language Problems and Language Planning*

Tarvainen, Kalevi (1985): *Kontrastive Syntax Deutsch - Finnisch.*
Heidelberg: Groos

Tarvainen, Kalevi (1987): Formale Dependenzgrammatik des Finnischen.
Opublicerad rapport. Utrecht: BSO/Research

Tesnière, Lucien (1959/1982): *Éléments de syntaxe structurale.*
Paris: Klincksieck (2:a uppl., 4:e tryck. 1982)

Witkam, A. P. M. (1983): *Distributed Language Translation. Feasibility study of a multilingual facility for videotex information networks.*
Utrecht: BSO

Tove Fjeldvig
Statens Datasentral A/S
Oslo.

Anne Golden
Institutt for norsk som fremmedspråk
Universitetet i Oslo

BRUK AV SPRÅKBASERTE HJELPEMIDLER I INFORMASJONSSØKING

1. Dagens informasjonssøkesystemer

Med et informasjonssøkesystem sikter vi her til et system som kan håndtere uformaterte, tekstlige dokumenter. At det også kan håndtere strukturert, feltorganisert informasjon, er mindre interessant for denne sammenhengen.

Gjenfinning av dokumenter er basert på ordene i dokumentene, og i prinsippet kan alle ord anvendes som søkeord. Dette gir muligheten til å stille fleksible søkeargumenter, og det finnes ingen grenser for hva man kan søke på. Resultatet vil avhenge av hvilke dokumenter som inneholder disse søkeordene.

Samtidig stiller denne form for tekstsøking store krav til valget av søkeord. Skal man finne fram til et relevant dokument, må søkeordene finnes blant de ordene forfatteren har brukt til å uttrykke det aktuelle søkebegrepet. Dette byr ofte på problemer fordi et begrep kan uttrykkes ved ulike ord.

Dagens informasjonssøkesystemer er ikke i stand til å likestille ord som innholdsmessig gir uttrykk for det samme i et dokument. Selv ikke ord som er bøyd eller avledet av samme rot, blir forbundet med hverandre. Søker man f.eks. på ordet *mord*, finner man ikke de relevante dokumentene hvor bare formen *mordet* er brukt.

Enkelte systemer gir riktignok muligheten til å kalle opp en synonymtesaurus ved formuleringen av søkeargumentet som kan hjelpe brukeren til å finne synonyme søkeuttrykk. Her kan man definere ulike semantiske relasjoner mellom ord - og man kan også likestille ord med felles rot. Grunnen til at denne type hjelpemiddel likevel er lite brukt, er kostnaden forbundet med etablering og vedlikehold av dem. De fleste databaser vokser over tid, og skal en tesaurus være ajour med databasen, må den oppdateres når databasen oppdaters. For mange er dette en nærmest uoverkommelig oppgave.

Det eneste hjelpemiddelet som er vanlig i dagens informasjonssøkesystemer, er trunkering. Dette er en primitiv liten algoritme som gjør det enkelt å utvide søkeargumentet med alle ord som innledes (ev. avsluttes) med en gitt

tegnstreng. Ved å trunkere et søkeord vil man kunne få fanget opp ord som er bøyd, avledet eller sammensatt av søkeordet. Man vil også kunne få med en del ikke-relevante ord, men dette trenger nødvendigvis ikke påvirke søkekvaliteten. Undersøkelser viser at problemet med trunkering er heller at mange brukere ikke gjør benytter seg av den. Enten glemmer de å trunkere, eller så har de ikke forstått hvor viktig det er (jfr. Fjeldvig 1987:43-52).

Vi har derfor stor tro på at en tilføring av nye hjelpemidler i dagens informasjonssøkesystemer som kan bistå brukeren i formuleringen av søkeargumentet, vil kunne øke søkekvaliteten for mange. Gjenfinningsgraden[1] vil kunne øke som en følge av at søkeargumentet blir supplert med flere adekvate søkeord. Likeledes vil presisjonen[2] kunne øke som følge av en bedre rangering av de funne dokumenter.[3]

2. Utvikling av språkbaserte hjelpemidler for tekstsøking

For å bøte på denne mangelen, ble det i 1980 satt i gang et prosjekt ved Institutt for Rettsinformatikk, Universitetet i Oslo, som bl.a. hadde til formål å utvikle metoder for automatisk rotlemmatisering og for automatisk gjenkjenning og splitting av sammensatte ord.

Metoden for automatisk rotlemmatisering skulle sørge for å likestille ord som var bøyd eller avledet av samme rot. Et slikt hjelpemiddel vil stille brukeren helt fritt til å velge formen på søkeordet, for systemet vil sørge for at alle bøynings- og avledningsformer kommer med.

På tilsvarende måte skulle metoden for automatisk splitting av sammensatte ord ta hånd om de sammensatte søkeordene og supplere disse med et uttrykk som også dekket eventuelle omskrivinger av disse, f.eks. *knivtrussel* vil bli splittet i *kniv* og *trussel* som vil dekke omskrivingen *"trussel med kniv"*.

I tillegg ønsket vi å se nærmere på muligheten for automatisk (høyre)trunkering. Ofte vil dette kunne være et bedre alternativ enn automatisk rotlemmatisering, bl.a. fordi en trunkering også fanger opp ord som er sammensatt av søkeordet, f.eks. *mord** (hvor * er brukt om trunkeringssymbol) dekker ordene *mordvåpen*, *mordoffer* og *mordkveld*.

---

1. Gjenfinningsgraden (eng. recall) er et uttrykk for hvor stor andel av de relevante dokumentene som er funnet i et søk.

2. Presisjonen er et uttrykk for hvor stor andel av de funne dokumentene som er relevante.

3. Formålet med en rangering er presentere først for brukeren de dokumentene som har størst sannsynlighet for å være relevante. I slike tilfeller vil rangeringskriteriet ofte være basert på den totale søkeordfrekvensen. Dette bygger på en hypotese om at jo flere ganger et ord er nevnt i teksten, jo større sjanse er det for at ordet reflekterer innholdet i teksten. Hvis man her utelater aktuelle søkeord, f.eks. angir mord som søkeord, men ikke mordet, vil man kunne få plassert et dokument som inneholder ordet mord tre ganger foran ett som inneholder mord to ganger og mordet 20 ganger.

Det var en forutsetning at metodene skulle baseres på et sett med regler - og ikke en forhåndsdefinert ordbok. Dette er viktig, fordi de fleste institusjoner som anskaffer et informasjonssøkesystem av denne type, har store - og ofte voksende - databaser. Med en regelbasert metode mener vi her en metode som er basert på et sett med regler som inneholder generell informasjon om ordenes bøynings- og avlednings-muligheter. Dette vil være en språkavhengig metode, men den vil ikke være avhengig av den enkelte database.

3. Kort om metoden for automatisk rotlemmatisering

Metoden for automatisk rotlemmatisering grupperer alle ord som tilhører samme rotlemma [4] ved å gi disse ordene en felles oppslagsform. Oppslagsformen kommer fram ved at de enkelte grafordene sammenliknes bakfra med de ulike bokstavstrenger på en regelliste. Hvis grafordet helt eller delvis overlapper bokstavstrengen og alle eventuelle betingelsene oppfylles, utføres visse ordrer. Den vanligste ordren er at de bokstavene som utgjør en endelse, skal strykes, men den kan også være at ordet skal behandles på nytt, eller at visse bokstaver skal legges til.

Oppgavene man står ovenfor ved rotlemmatisering av norske graford, kan føres tilbake til følgende tre punkter, identifisere bøyningsendelser, eks. *-en* i *arven*, identifisere avledningsendelser, eks. *-ing* i *arving*, sammenføre røtter som er realisert som ulike rotformer, eks. *far* og *fedre*.

Eksempler: Grafordene *sykkel, sykkelen, syklene* blir behandlet fordi strengene KKEL, ELEN, og ENE finnes på regellista. Alle disse strengene stiller som betingelse at det står noe foran strengen. Ordrene er forskjellige for disse reglene:

ENE - fjern 3 tegn
KKEL - fjern 4 tegn, legg til tegnet L
ELEN - fjern 2 tegn, så ny behandling

Denne behandlingen fører til at alle *sykkel, sykkelen, syklene* får oppslagsformen SYKL.

En utførlig beskrivelse av metode for automatisk rotlemmatisering er gitt i Fjeldvig/Golden 1986 og Fjeldvig 1987:65-98.

Reglene
Regellista gir en oversikt over bøyningsendelsene og avledningsendelsene, dvs. de bundne morfemene i norsk. I tillegg gir den en oversikt over strenger som blir sammendratt ved bøyning og en del ord som har uregelmessig bøyning så sant disse ordene kan avgrenses til lukkede grupper. Tilsammen er det 684 regler på regellista.

---

4. Et rotlemma er en samling av alle ord som kan føres tilbake til samme rot uten at betydningen til ordet blir endret. Oftest er altså et rotlemma det samme som en stamme uten avledningsendelser. I noen tilfeller har imidlertid avledningsendelsene ført til at den nye stammen er blitt leksikalisert, dvs fått en spesiell betydning i forhold til grunnordet. Da utgjør den nye stammen et eget rotlemma.

Resultat
Metoden ble testet på et tekstkorpus som var satt sammen av tekster fra juridisk materiale og skolebøker i fysikk, geografi og historie. Tekstkorpuset inneholdt ca 1/2 millioner løpende ord og i overkant av 23000 graford. Resultatet viste at 97,7 % av alle grafordene fikk en oppslagsform som førte til at det kom i riktig rotlemma.

Feilenes betydning for informasjonssøking
2,3 % av alle grafordene ble ikke samlet i ett og bare ett entydig rotlemma. Det var tre feiltyper som var mulige. Feiltype a) besto i at grafordene som egentlig tilhørte samme rotlemma, ble delt i to eller flere grupper. Det var 1,4% av grafordene som ble feilplassert på denne måten. Hvis ett av disse grafordene ble valgt som søkeord, ville ikke søkeargumentet bli utvidet med alle de andre bøynings- og avledningsformene til ordet. Denne feilen kan altså føre til at man ikke finner alle de relevante dokumentene, dvs. gjenfinningsgraden kan altså bli dårligere.

Feiltype b) besto i at grafordene som tilhørte ulike rotlemmaer, ble slått sammen til ett rotlemma. Det var 0,8% av grafordene som ble feilplassert på denne måten. Hvis ett av disse grafordene ble brukt som søkeord, ville søkeargumentet bli utvidet med alle bøynings- og avledningsformene til ordet, men også andre irrelevante ord. Denne feilen kan altså føre til at man finner flere irrelevante dokumenter, dvs. presisjonen blir dårligere.

Feiltype c) besto i at grafordene som egentlig tilhørte samme rotlemma fordelte seg på flere grupper som også inneholdt andre rotlemma. Det var 0,2% av grafordene som ble feilplassert på denne måten. Hvis ett av disse grafordene ble brukt som søkeord, vil ikke søkeargumentet bli utvidet med alle bøynings- og avledningsendelsene til ordet, men derimot kan det bli utvidet med andre irrelevante søkeord. Både gjenfinningsgraden og presisjonen kan altså bli dårligere.

4. Kort om metoden for automatisk trunkering

Den automatiske trunkeringen ble utviklet som et alternativ til den automatiske rotlemmatiseringen. Dette er et nyttig alternativ i tilfeller hvor dokumentbasen ikke er tilgjengelig for bearbeiding for soking. Egentlig dreier ikke dette seg om en egen metode, men snarere en annen anvendelse av rotlemmatiseringen i informasjonssøking. Systemet står selv for trunkeringen ved først å rotlemmatisere søkeordet og så trunkere både oppslagsformen og søkeordet. I de fleste tilfeller er oppslagsformen identisk med grunnformen, slik som oppslagsformen *arv*. I tilfeller hvor det forekommer ulike rotformer innen et paradigme, har vi gitt rotlemmaet den synkoperte rotformen som oppslagsform, f.eks vil det rotlemmaet som *regel* (rotform: *regel*) og *regler* (rotform: *regl*) tilhører, ha oppslagsformen *regl*. Den automatiske trunkeringer vil imidlertid sørger for at begge rotformene blir trunkert.

En nærmere beskrivelse av metode for automatisk trunkering er gitt i Fjeldvig 1987:149-170.

5. Kort om metoden for automatisk gjenkjenning og splitting av sammensatte ord.

Denne metoden skiller først ut de enstavede usammensatte ordene (dvs. ord med en stavelse i roten), for så å identifisere de ulike morfemene i de resterende ordene.

Et hvert norsk graford ***(O)*** vil passe inn i formelen (1):

(1) ***O: pre* rot avl* ((E/S) pre* rot avl*)* bøyn****

hvor

pre* står for 0, 1 eller flere prefikser
rot står for rot
avl* står for 0, 1 eller flere avledningsendelser
(E/S) står for mulig fuge-E eller fuge-S
bøyn* står for 0, 1 eller flere bøyningsendelser
( )* står for 0,1 eller flere forekomster av det som står inni parantesen

Stavelsesstrukturen i norsk kan beskrives ved hjelp av formel (2) som viser strukturen i enhver morf ***(M)***:

(2) ***M: ini vok (med vok)* fin***

hvor

ini betyr initialkluster dvs. 0, 1 eller flere konsonanter som forekommer initialt i morfen
med betyr medialkluster dvs. 0, 1 eller flere konsonanter som forekommer medialt i morfen
fin betyr finalkluster dvs. 0, 1 eller flere konsonanter som forekommer finalt i morfen

De bundne morfemene, dvs prefiksene, avledningsendelsene og bøyningsendelsene finnes bare i et begrenset antall, og det er svært sjelden at det kommer nye medlemmer inn i disse gruppene slik at vi kan regne dem for lukkede. Disse morfene har vi derfor oversikt over. Det samme gjelder de ulike konsonantklustrene.

I prinsippet består oppgaven i først å finne fram til alle mulige morfemgrenser ut fra formel (2), for så å redusere dette løsningsforslaget ut fra formel (1). Deretter rangeres de ulike forslagene.

En mer utforlig beskrivelse av denne metode er gitt i Fjeldvig 1987:99-148 og Fjeldvig/Golden 1987.

Reglene
Ved utviklingen av denne metoden tok vi utgangspunktet i den samme oversikten over de bundne morfemene som vi brukte til metoden for automatisk rotlemmatisering. Men informasjonen, betingelsene, og kravene er annerledes på denne regellista. I tillegg gjør vi bruk av en liste over de ulike konsonantklustrene og deres plasseringsmuligheter i en morf. Framgangsmåten går ut på å sammenlikne bokstavstrengene i grafordet med disse regellistene.

Resultat
For å teste hvor vellykket metoden var når det gjaldt å kjenne igjen sammensetninger, ble et tilfeldig utvalg på 1019 graford testet, 149 av disse var sammensatte. 1018 av grafordene ble riktig vurdert. Den ene feilen var et usammensatt graford som ble behandlet som et sammensatt. Resultatet var altså tilnærmet lik 100% riktig.

Når det gjaldt splittingen, ble den testet på et tilfeldig utvalg på 160 graford. Her ga metoden alternative løsninger som ble rangert. For 144 av de sammensatte grafordene (90%) kom den riktige løsningen på 1. plass. I 6 tilfeller kom den riktige løsningen på delt 1. plass, i 6 tilfeller på 2. plass, i 1 tilfelle på 3. plass, i 2 tilfeller på 4. plass og i 1 tilfelle på 5. plass. I 97,5% kom altså den riktige løsningen på 2. plass eller bedre.

Feilenes betydning for tekstsøking
I vårt lille eksperimentmateriale ble alltid den riktige sammensetningsgrensen funnet, selv om denne løsningen i 10% av tilfellene ble rangert lavere enn andre uriktige forslag. Selv om disse ordene ble brukt som søkeord, vil ikke dette føre til noe problem, siden én feil deling vil gi ord som ikke eksisterer i norsk. Man kan da først undersøke om den delingen man får, inneholder ord som forekommer i databasen. Hvis de ikke gjør det, kan man la systemet gå videre med neste forslag og undersøke om man får tilslag her på de enkelte leddene. Hvor mange forslagene man skal undersøke, er et spørsmål om hvor store ressurser man er villig til å bruke. I følge vår undersøkelse vil det være rimelig å stoppe etter de 2-3 første forslagene, fordi den riktige løsningen finnes som oftest bl.a. disse.

6. Forsøk med metodene i tekstsøking

Som et ledd i arbeidet med disse metodene, ble det gjennomført flere forsøk med formål å undersøke hvilken effekt de vil ha på søkeresultatet. Ett av dem gikk ut på å undersøke hvor mange flere dokumenter som ble funnet når man søkte på alle bøyning- og avledningsformer til et ord i stedet for bare grunnformen til ordet. (Det er grunnformen som oftest bli anvendt i søkeargumentet.) Forsøket omfatte 121 søkeord, og søkingen var rettet mot en juridisk database som omfattet ca. 14-15000 sammendrag av høyesteretts-avgjørelser (én av LOVDATA's databaser). Resultatet viste en gjennomsnittlig økning i antall funne dokumenter på hele 215%.

Et annet forsøk var rettet mot sammensatte søkeord, og tok sikte på å undersøke i hvor stor grad man var i stand til å fange opp omskrivinger av de sammensatte søkeordene ved å søke på uttrykk som besto av en kombinasjon av de enkelte leddene i det sammensatt ordet (eller bøynings- og avledningsformer av disse). Dette forsøket omfattet 54 sammensatte ord, og resultatet viste at i 49 av disse tilfellene ble omskrivinger fanget opp. Det ble stilt som krav at de enkelte (usammensatte) ordene skulle forekomme i samme setning.

Disse undersøkelsene viser at både metode for automatisk rotlemmatisering og metode for automatisk splitting av sammensatte ord vil kunne bidrar til mer fullstendige søkeargumenter. Derimot gir de ikke informasjon om den endelige effekten av disse metodene på søkeresultatet da dette vil avhenge av både

brukerens informasjonsbehov, den totale mengden med søkeord og søkestrategien, samt den aktuelle databasen. Det er samspillet mellom disse faktorene som er bestemmende for søkekvaliteten.

For derfor å få nærmere innsikt i effekten av bruk av disse metodene på søkeresultatet, ble det gjennomført en serie med kontrollerte forsøk i tekstsøking. Lignende forsøk med automatisk rotlemmatisering har vært gjennomført tidligere for engelsk (Salton 1968) og tysk (Niedermair/ Thurmair/Bütler 1984), og begge ga positive resultatet.

Beskrivelse av et kontrollert forsøk
Et kontrollert forsøk i tekstsøking går i korte trekk ut på at man med utgangspunkt i en gitt dokumentsamling og et sett med spørsmål, sammenligner resultatet av en maskinell søking med resultatet av en manuell gjennomlesing av hele dokumentsamlingen. Resultatet uttrykkes ofte ved bruk av effektivitetsmålene gjenfinningsgraden (G) og presisjon (P). Ved å dele inn resultatlista i rangsett - noe som forutsetter en søkestrategi som rangerer de funne dokumentene - kan resultatet av et søk presenteres i en GP-kurve. På bakgrunn av de individuelle GP-kurvene kan man så beregne en gjennomsnittlig GP-kurve som gir uttrykk for søkekvaliteten for dette søkesettet. Ved å gjennomføre flere slike forsøk for ulike typer søkeargumenter (eller søkestrategier), vil man ved å sammenligne de gjennomsnittlige GP-kurvene kunne si noe om hvilke type søkeargumenter som gir best resultat. Man vil så kunne gjennomføre en nærmere undersøkelse av de enkelte søk for å få innsikt i hvorfor resultatet er som det er.

Våre forsøk var basert på et eksperimentmateriale som omfattet ca. 1300 sammendrag av domsavgjørelser i familie-, skifte- og arverett.[5] Det ble utformet 22 spørsmål av varierende kompleksitet og lengde. En mer utførlig beskrivelse av eksperimentmaterialet og forsøkene er gitt i Fjeldvig 1987:171-200.

Den manuelle gjennomlesingen av dokumentsamlingen ble foretatt av de samme personene som hadde stilt spørsmålet. Hvert spørsmål ble behandlet for seg, og de relevante dokumentene ble notert på en liste. Denne listen kalles ofte fasiten til spørsmålet.

For hvert spørsmål ble det konstruert et søkeargument som bare omfattet ord som forekom i spørsmålet. Det var gjennomsnittlig 5 søkeord pr. søkeargument. Ved søking ble alle dokumenter som inneholdt minst ett av disse søkeordene, valgt ut. Deretter ble de funne dokumentene rangert ut fra hvor mange ulike søkeord de inneholdt.

Det første forsøket ble gjennomført med søkeargumenter som bare inneholdt grunnformen til søkeordene. Den gjennomsnittlige GP-kurven som dette ga, er representert ved den heltrukne kurven i fig. 1. Denne kurven gir altså uttrykk for de søkeresultatene vi fikk uten å ta i bruk noen av våre metoder.

Eksempel: Ett av våre spørsmål lød omtrent som følger: *Har et barns handicap betydning for oppfostringsbidragets størrelse?* Søkeargumentet besto i dette tilfellet av følgende ord:

---

5. Totalt antall ord var ca. 190 000 og antall ulike graford ca. 11 000.

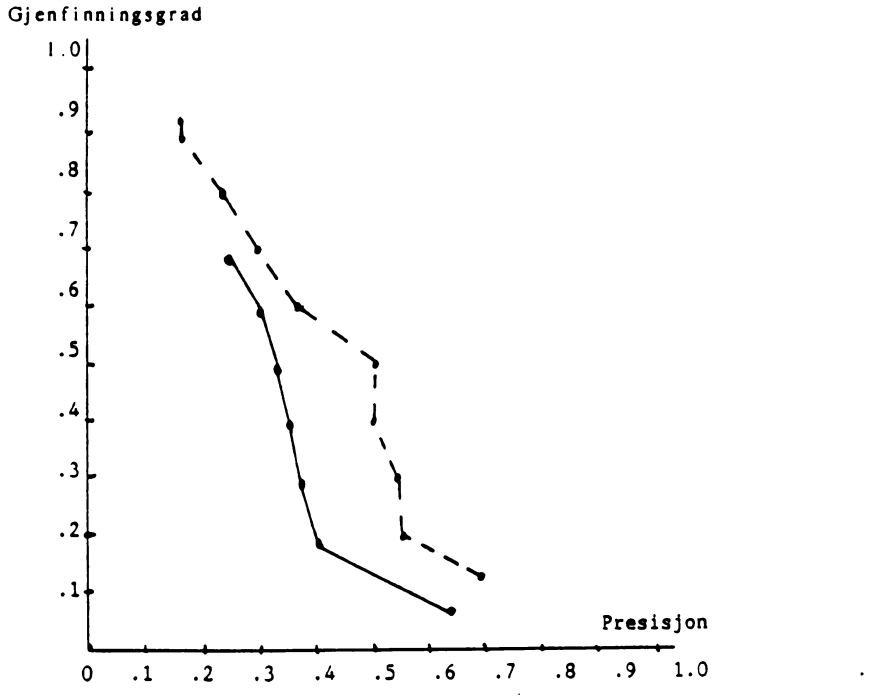
barn
handicap
oppfostringsbidrag

Automatisk rotlemmatisering
Deretter erstattet vi hvert søkeord med en gruppe med ord som inneholdt alle de bøynings- og avledningsformene til dette ordet som forekom i databasen. Søkeargumentet i eksempelet ovenfor omfattet nå følgende ord:

(barn, barns, barnet, barnets, barna, barnas)
(handicap, handicapet, handicapede, handicapedes)
(oppfostringsbidrag, oppfostringsbidraget, oppfostringsbidragets)

Rangeringen av de funne dokumentene ble nå foretatt på bakgrunn av hvor mange slike grupper som var representert i dokumentet. Resultatet er representert ved den stipplete kurven i fig. 1.

Fig. 1 Effekten av å utvide søkeargumentene med bøynings- og avledningsformer



-------- den gj.snitt. GP-kurven før utvidelsen av søkeargumentene
- - - - den gj.snitt. GP-kurven ved bruk av automatisk rotlemmatisering

Sammenligner vi nå disse to kurvene, ser vi at vi har oppnådd et betydelig bedre resultat med bruk av metoden for automatisk rotlemmatisering. En sammenligning av resultatene for hvert enkelt søk viste også en forbedring av søkekvaliteten i over halvparten av tilfellene. Den gjennomsnittlige økningen i gjenfinningsgraden var på hele 24%.

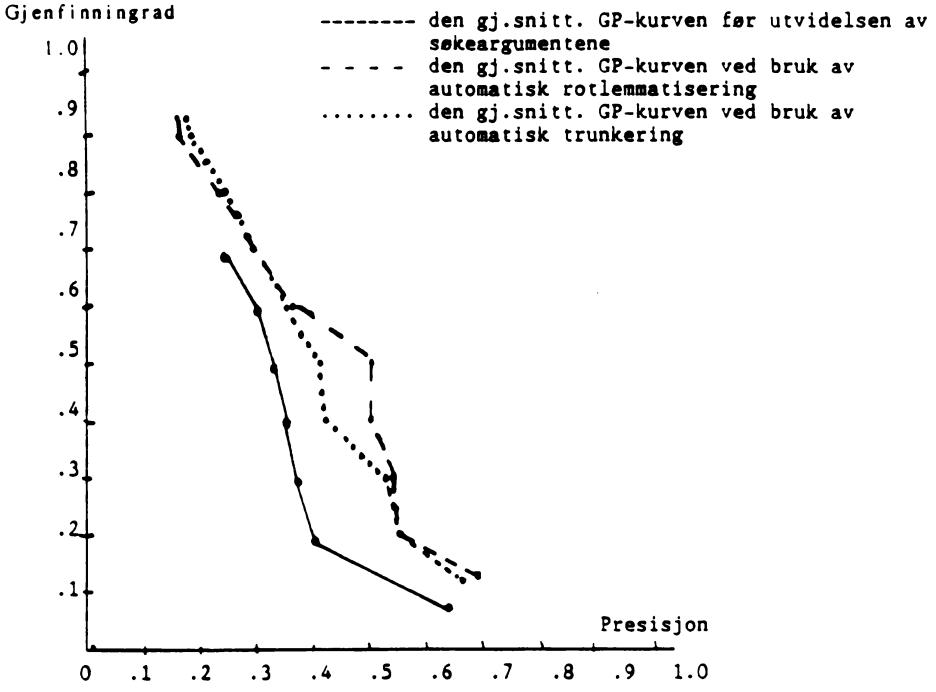
Automatisk trunkering
På lignende måte ble det gjennomført et kontrollert forsøk med automatisk trunkering. Her ble hvert (opprinnelig) søkeord automatisk trunkert.

Eksempel:

*barn**
*handicap**
*oppfostringsbidrag**

Resultatet er gjengitt ved den prikkete linjen i fig. 2. nedenfor. De to øvrige kurvene i figuren er de samme som i fig. 1.

Fig. 2 Sammenligning av resultatet ved automatisk trunkering og automatisk rotlemmatisering



En sammenligning av kurven for automatisk trunkering med den vi fikk ved eksplisitt å supplere søkeargumentet med alle bøynings- og avledningsformer.

viser helt klart at den automatisk trunkering absolutt er et aktuelt hjelpemiddel ved tekstsoking. Forskjellen mellom disse to kurvene er helt marginal, og det er vanskelig å si om den ene kurven er bedre enn den andre.

En nærmere undersokte av de enkelte søkene viste at den automatiske trunkeringen i et par av tilfellene hadde ført til at flere relevante dokumenter. De fleste resultatlistene inneholdt nå flere irrelevante dokumenter, men disse dokumentene havnet som oftest lenger ned på lista fordi de inneholdt relativt få søkeord.

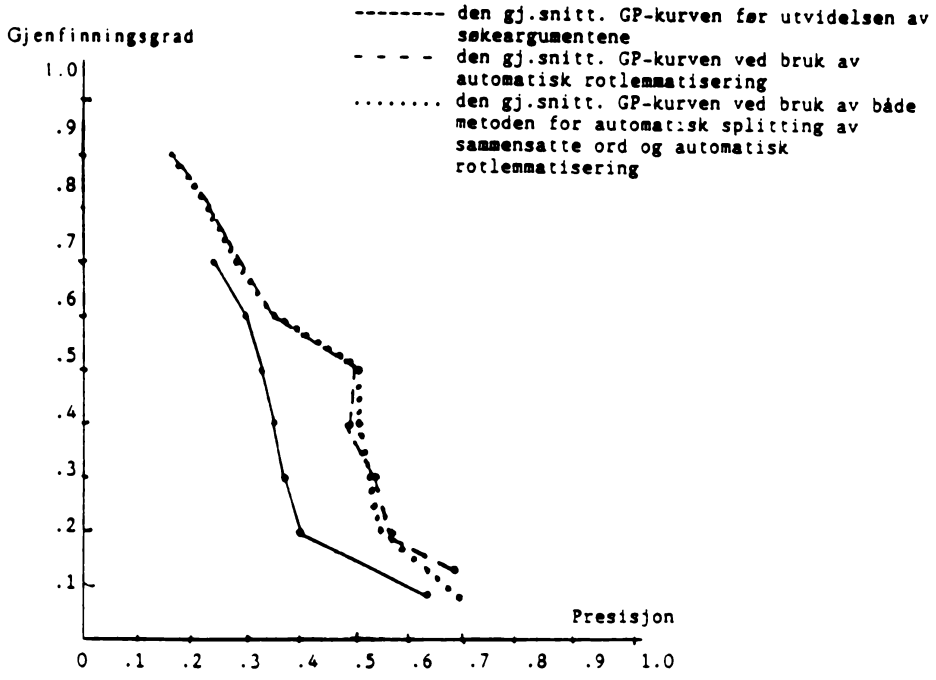
Automatisk gjenkjenning og splitting av sammensatte søkeord.
I neste forsøk ble i tillegg alle sammensatte søkeord splittet og supplert med et uttrykk som inneholdt de enkelte leddene i det sammensatte ordet, f.eks. ordet *oppfostringsbidrag* ble supplert med uttrykket

(oppfostre, oppfostrer, oppfostret, oppfostring, oppfostringen)
SETN (*bidra, bidrar, bidratt, bidraget, bidragets, bidragene, bidragene)*

Operatoren SETN, også kalt setningsoperatoren, stiller som krav at ett av ordene i den første parantesen må forekomme i samme setning som ett av ordene i den andre parantesen.

Resultatet av dette forsøket er gjengitt ved den prikkete kurven i fig. 3. Den stipplete kurven er den samme som i fig. 2, dvs. resultatet uten at de sammensatte søkeordene er behandlet.

Fig. 3 Effekten av å splitte de sammensatte søkeordene



Som det fremgår av figuren, ga denne utvidelsen av søkeargumentet lite utslag på det gjennomsnittlige søkeresultatet. Den førte ikke til at det ble funnet flere relevante dokumenter, fordi de fleste relevante dokumentene var alt funnet på bakgrunn av de øvrige søkeordene. Rangeringen ble bedre i noen av tilfellene, men i de fleste tilfellene var økning i antall søkeord i de relevante dokumentene for liten til å kunne endre rekkefølgen i resultatlista.

Jevnt over førte denne utvidelsen til at det ble funnet flere irrelevante dokumenter. Dette hadde heller ikke særlig innflytelse på resultatet fordi disse dokumentene inneholdt relativt få søkeord og ble derfor plassert i de bakerste rangsett.

En nærmere undersøkelse av hvert enkelt søk viste imidlertid at denne behandlingen av de sammensatte søkeordene i ca. halparten av tilfellene hadde ført til at søkeargumentet nå inneholdt søkeuttrykk som fanget opp omskrivinger av de sammensatte søkeordene. I de resterende tilfellene utgjorde de sammensatte ordene typiske juridiske faguttrykk (f.eks. *ektepakt* og *halvpart*) eller leksikaliserte ord (f.eks. *pleiedatter* og *lønnsbringende*). Det var en langt høyere andel av denne type ord enn det vi hadde oppnådd i andre sammenhenger. Likevel er det sjelden at en splitting av denne type ord vil få negativ effekt på søkeresultatet, fordi de enkelte usammensatte ordene ikke vil forekomme i samme setning.

7. Konklusjon

Resultatene av disse forsøkene viser helt klart at både metoden for automatisk rotlemmatisering og metoden for automatisk trunkering gir en betydelig forbedring av søkekvaliteten. Hvilke av de to fremgangsmåtene man bør satse på, vil avhenge av ulike hensyn både av systemmessig og ressursmessig art.

Forsøket med automatisk splitting av sammensatte ord viste at dette hadde liten innflytelse på søkekvaliteten. Likevel har vi tro også på en slik metode i informasjonssøking, fordi forsøket tross alt viste at vi på denne måten får utvidet søkeargumentene med adekvate søkeuttrykk. At resultatet ikke ble bedre i dette forsøket, mener vi skyldes ulike egenskaper ved eksperiment-materialet.

Tatt i betraktning at svært mange brukere (typisk uøvde og sporadiske brukere) verken trunkerer søkeordene eller på annen måte sørger for å få med de ulike bøynings- og avledningsformene til søkeordene (jfr. Fjeldvig 1987:53-52), anser vi disse metodene for å være et vesentlig bidrag til løsningen av synonymproblemet i tekstsøking.

LITTERATURLISTE

Fjeldvig, Tove/Golden, Anne (1984) *Automatisk rotlemmatisering - et lingvistisk hjelpemiddel for tekstsøking.* CompLex 9/84, Universitetsforlaget. Oslo.

Fjeldvig, Tove (1986) *Tekstsøking - teori, metoder og systemer.* Universitetsforlaget. Oslo.

Fjeldvig, Tove/Golden, Anne (1986) "Automatisk splitting av sammensatte ord - et lingvistisk hjelpemiddel for tekstsøking"; Karlsson 1986:73-82.

Fjeldvig, Tove (1987) *Effektivisering av tekstsøkesystemer. Utvikling av språkbaserte metoder.* Universitetsforlaget. CompLex nr. 13/87. Oslo.

Gavare, Rolf (1979) "Automatisk lemmatisering utan stamlexikon - Några synspunkter tio år efteråt". Maegaard 1979: 123-131.

Hellberg, Steffan (1971) "Automatisk lemmatisering - En modell for upprättande av böyningsserier i ett frekvenslexikon". Språkdata. Gøteborg.

Karlsson, Fred (ed) (1986) *Papers from the 5th Scandinavian Conference of Computation and Linguistics.* University of Helsinki, Department of General Linguistics.

Källgren, Gunnel (1985) *En algoritme för deling av sammensatte ord i svenskan.* Institutionen för lingvistik. Stockholms Universitet. Stockholm.

Maegaard, Bente (1979) *Nordiske datalingvistikdage i København 6.-10. oktober 1979.* Foredrag. Institut for anvendt og matematisk lingvistik, Københavns Universitet 1979. København.

Munthe, Synneve Kjuus Munthe (1972) *Sammensatte ord. En kvantitativ undersøkelse av norsk litteratur og sakprosa.* Hovedfagsoppgave ved Nordisk institutt, Universitetet i Bergen og Oslo.

Niedermair, G.T./Thurmair G./Büttel I. (1984) "MARS - a retrieval tool on the basis of morphological analysis", van Rijsbergen 1984:369-382.

Salton, Gerard (1968) *Automatic Information Organization and Retrieval.* McGraw-Hill computer series.

van Rijsbergen C.J. (1984) *Research and Development in Information Retrieval.* Proceedings of the third joint BCS ans ACM symposium King's College, Cambridge 2.6 July 1984. British Computer Society Workshop Series.

Lennart Lönngren, Uppsala

LEXIKA, BASERADE PÅ SEMANTISKA RELATIONER

Den första fråga man måste ta ställning till om man vill bygga upp en tesaurus, alltså ett lexikon baserat på semantiska relationer, är om man skall tillämpa någon form av hierarkisering och hur i så fall denna skall se ut. I princip vill jag förkasta tanken på att begreppen som sådana kan ordnas hierarkiskt; jag tror alltså inte på några universella eller särspråkliga semantiska primitiver à la Wierzbicka (1972). Normalt är det nog att konstatera att det föreligger ett associativt samband mellan två begrepp, t.ex. *tand* och *bita*, samt att fastställa styrkan hos och arten av detta samband utan att postulera något riktningsförhållande.

Det är emellertid praktiskt att organisera ett tesauruslexikon hierarkiskt. Det innebär en förenkling så till vida att man ersätter en mångfald av relationer med i princip en enda, dependens. Jag tänker mig här en mer djup- och genomgående hierarkisering än den vi finner i Rogets lexikon (1962, första gången utgivet 1852) och dess svenska efterbildning, Bring (1930), där man definierat ett begränsat antal "begreppsklasser" och hänfört varje ord till en sådan. Frågan är bara om detta är möjligt, alltså om en sådan hierarkisering står i samklang med ordskattens inre natur. För att citera Kassabov (1987, 51) gäller det här att undvika det allmänna misstaget att "attempt to prove the systematic character of vocabulary not by establishing the inherent principles of its inner organization, but by forcing upon the lexical items the networks of pre-formulated systems".

Om vi börjar med den del av ordförrådet som är sammanbunden morfologiskt-derivationellt, alltså det subsystem som omfattas av ordbildningsrelationer, så är det ju helt uppenbart att språktecknens uttryckssida kännetecknas av skiftande komplexitet och att man kan se ett slags vertikal relation, alltså riktnings- eller dependensförhållande, mellan lexem som *sand* och *sandig*, där det ena är avlett av det andra. Att organisera ett språks hela ordförråd i hierarkiska ordbildningsnästen är dock ingen lätt sak. En orsak till detta är att lexemens betydelser långt ifrån alltid återspeglas av ordbildningsstrukturen, en annan att det finns så många fall av identisk komplexitet, där leden trots detta måste ordnas i vertikalled, därför att det inte finns något mindre komplext lexem i ordbildningsnästet i fråga, t.ex. *såg - såga, ljus - lysa*.

I de fall uttryckssidans komplexitetsförhållanden förefaller klara har man i allmänhet antagit att det finns en direkt komplexitetsmotsvarighet på innehållssidan, alltså att i vårt exempel *sandig* är även semantiskt mer komplext än *sand* och innehåller komponenten 'sand' på samma sätt som strängen *sandig* innehåller strängen *sand*. Detta resultat kommer man ju också till om man förutsätter att vi har att göra med tre språktecken (förutom flexionsmorfemet) - vardera med en uttrycks- och en innehållssida - nämligen de elementära *sand-* och *-ig* samt det komplexa *sandig*. Men då får man problem i sådana fall där den semantiska relationen förefaller identisk men där den formella komplexiteten pekar åt olika håll, t.ex. i paren *undersöka - undersökning* och *analys - analysera*, *fysik - fysiker* och *biolog - biologi*. Vill man fastställa språkoberoende semantiska hierarkiska förhållanden får man ytterligare problem i fall där nära semantiska ekvivalenter i två språk uppvisar olika riktningsförhållanden med avseende på formell komplexitet, jfr sv. *seger - segra* -

segrare men tj. *vítěz* ('segrare') - *vítězit* ('segra') - *vítězství* ('seger'). Det förefaller ohållbart att försöka lösa alla dessa problem genom att laborera med t.ex. nollsuffix, tomma suffix och tilläggskriterier av mer eller mindre semantisk natur. En språkoberoende hierarkisering försvåras naturligtvis också av att det ofta förekommer att två begrepp är sammankopplade ordbildningsmässigt i ett språk men inte i ett annat, t.ex. *nyckel*, som är (synkroniskt) oderiverat i svenska och engelska, men i tyska och finska deriverat av 'låsa' (*Schlüssel*) resp. 'öppna' (*avain*); vidare är det vanligt att en och samma företeelse har olika derivationell anknytning, jfr t.ex. 'näsduk', som i svenska och ryska är förknippat med 'näsa', i tjeckiska, tyska och danska med 'ficka', i engelska med 'hand' etc.

Det är ändå viktigt att framhålla, att problemen visserligen är många nog att förbittra tillvaron för den som vill sammanställa ett hierarkiskt derivationslexikon, men ändå inte så många att de fördunklar det faktum att det finns en tendens, särskilt inom ramen för ett språksystem men även universellt: formell komplexitet tycks harmoniera med vissa andra egenskaper, framför allt med allmän frekvens (dvs hur ofta använt och därmed hur väl känt ett ord är för den genomsnittlige språkbäraren). Det är inte bara så att i varje frekvensordlista ordlängd och frekvens är i stort sett omvänt proportionella utan det förhåller sig också så att av två ord som är direkt "vertikalt" förbundna med varandra i en ordbildningsrelation är det enklare, alltså dominerande, lexemet oftast mer frekvent än det deriverade.

Mot denna bakgrund förefaller det ovannämnda företaget inte helt utsiktslöst. Det är emellertid, som alltid, viktigt att börja med tydliga fall och med dessa som mönster fortskrida till de mindre tydliga. En lämplig utgångspunkt är reguljära sammansättningar av typen *metallindustri* där hela ordet utgör en hyponym till huvudledet *industri*; sammansättningen är också mindre frekvent än huvudledet. Man kan alltså säga att dessa egenskaper klart harmonierar med den derivationella strukturen.

Sådana idealiska förhållanden råder långt ifrån alltid i sammansättningar. Dessutom måste vi kunna hantera andra typer av morfologiska samband samt fall där derivationellt samband saknas, t.ex. i par av typen *tand* : *bita*. Vi måste också ofta acceptera att ett dominerande lexem inte alltid kan vara en hyperonym. För det första är det svårt att hitta hyperonymer till många lexem, t.ex. *vanlig*, *skratta*, *dimma*, för det andra är hyperonymen inte alltid den starkaste, mest markanta relationspartnern. Det ger t.ex. inte så mycket att ge *del* som dominerande lexem till *kroppsdel* och *världsdel*; här är det i stället bättre att ge 'helheten', dvs *kropp* resp *värld*. Likaså är det troligen bättre att koppla samman *järnväg* med *tåg* än att låta det domineras av lexemet *väg*; och kopplingen till metallen *järn* är klart perifer, vilket visar att man inte alltid kan låta sig vägledas av ordbildningsstrukturen; ett annat tydligt exempel på detta är *skärgård*, som inte bör analyseras i sina sammansättningsled utan kopplas ihop med lexemet *ö*, jfr här t.ex. finskans *saaristo*. Genom att på detta sätt frigöra oss från ordens inre form tar vi ett steg på vägen mot en universell begreppshierarkisering.

Vad vi främst behöver är ett oberoende semantiskt kriterium som är mer generellt än "hyponymi-hyperonymi"-relationen, en assymmetri som kan läggas till grund för ett dependensförhållande. Det finns faktiskt en sådan assymmetri, och jag skulle vilja kalla den "lexikalisk determinism". Den består i att man givet en semantisk relation och en relationspartner A med större entydighet kan bestämma relationspartnern B än omvänt, dvs än vad som blir fallet om man

utgår från B och söker A. De tydligaste exemplen återfinner vi inom hyponymi-hyperonymi-relationen samt inom del-helhetsrelationen. Sålunda kan vi gå entydigt från *bil* till *fordon*, från *röd* till *färg*, från *styre* till *cykel*, från *nav* till *hjul* osv., men inte omvänt. Från dessa enkla fall kan vi gå vidare till andra semantiska relationer. Det finns också någonting som vi kan kalla "situationell determinism", nämligen i fall som *snyta* ← *näsa* (det semantiska objektet för 'snyta' är alltid 'näsa', medan däremot 'näsa' kan förekomma i många andra situationer än 'snyta'; ett likartat förhållande råder mellan *regna* och *vatten*. Även inom ett ordbildningspar som *sten* → *stena* finner vi sådan assymmetri. Naturligtvis finns det många nära relaterade ordpar mellan vilka determinism saknas, t.ex. *cykel* och *hjul*: hjul förekommer ju inte endast på cyklar och cyklar har ju andra delar vid sidan av hjulen (även om hjulen är mycket väsentliga, vilket reflekteras av att benämningarna på *cykel* i många språk är deriverade av *hjul*). Ett sådant faktum som att alla hjul inte har ekrar behöver dock inte sänka determinismen mellan *hjul* och *ekrar*, eftersom ekrar ändå bara förekommer på hjul.

Jag har alltså accepterat ett dependensbegrepp som utgör en sammanvägning av flera olika faktorer, vilka på det hela taget samverkar, nämligen: a) formell komplexitet, t.ex. *sand* → *sandig*; b) frekvens, t.ex. *annars* → *eljest*; c) stilistisk markering (dvs ett stilistiskt neutralt lexem betraktas som dominerande), t.ex. *rolig* → *kul*, *båda* → *bägge*; d) determinism, t.ex. *näsa* → *snyta*, *hjul* → *nav*. Den bästa sammanfattande benämningen på detta dependensbegrepp torde vara "enkelhet", inbegripande såväl uttrycks- som innehållssida: av två lexem, mellan vilka man vill etablera ett direkt vertikalt samband, väljs det enklare som det dominerande.

Nu kan man invända: att återföra ett ord på enklare begrepp är inte något nytt. Det är vad man alltid har gjort i ordboksdefinitioner. Jag skulle vilja säga, att det är vad man tror sig ha gjort. I själva verket vimlar det av exempel på motsatsen. Särskilt vanligt är att mycket enkla begrepp definieras med hjälp av mer komplexa (mindre kända). Ett belysande exempel är Ralph (1979), där enkla verb av typen *jubla* jämförs med parafraser av typen *uttrycka jubel*. Andra exempel är *förvärva fångst* (= *fånga*), *konsumera dryck* (= *dricka*). Det är svårt att hålla med författaren: "Varje [i parafrasen] ingående ord blir följaktligen semantiskt mindre komplext än utgångsverbet." (s. 164). I jakten efter mer generella begrepp tvingas man ofta söka sig till språkets periferi, vilket f.ö. också Brings tesaurus är ett gott exempel på; här finner vi begreppsklasser som "Omängdhet" och "Inrymningsplats". Det är under sådana förhållanden inte underligt att de ofta påtalade cirkeldefinitionerna i enspråkiga lexika uppstår. Det är enligt min mening helt omöjligt att definiera varje lexem, särskilt lexemen i de övre frekvensområdena, med hjälp av (genuint) enklare ord. I den hierarkiska struktur jag här beskriver är det ju inte heller fråga om att definiera lexemen.

Med utgångspunkten i deterministiska sammansättningar bör man vid sidan av ett huvudled kunna acceptera även ett modifierande led som dominerande lexem, även om detta naturligtvis komplicerar lexikonets struktur. Vid ett lexem som *metallindustri* ansätts alltså två dominerande lexem, nämligen "föräldraparet" *industri* och *metall*; härvid kan vi kalla huvudkomponenten *industri* "moder" och den modifierande komponenten *metall* "fader" (komponenterna ges också alltid i denna ordning). Det är viktigt att observera att de ovan nämnda "harmonierande" egenskaperna vad gäller frekvens och determinism huvudsakligen endast gäller för modersdominanten, ej för fadersdomi-

nanten; denna utgör ju inte heller en hyperonym till derivatet.

För att fortsätta analogin med strukturen hos en familj är det endast vissa lexem som får ett komplett föräldrapar, alltså två dominanter; vid bildningar medelst affix är det mestadels tillräckligt att ange endast en förälder, t ex *sandig → sand*. När den morfologiska kopplingen saknas kan vi mera fritt välja mellan att ge en eller två dominanter; sålunda kan man tveka om man skall låta *öga* definieras enbart genom *se* eller genom *se* plus *ansikte*. Liksom det inom ordbildningen förekommer kopulativa sammansättningar, av typen *rödgul*, måste man även utanför den morfologiska sfären ibland ge föräldrapar som är kopulativt sammanbundna, t.ex. *ranglig ← lång + mager*.

Ett dominerande lexem kan naturligtvis ha många barn, i förhållande till vilket det kan uppträda som moder eller fader eller ensamförälder. Mera ovanligt är att ett och samma föräldrapar har flera barn, t ex *veta + vilja → intresse, undersöka, nyfiken...*

För att pröva om dessa tankar är realiserbara har jag hittills dels bearbetat en mindre korpus på 17.000 löpord och ur den skapat ett litet lexikon, dels kompletterat detta med ord ur allmänna frekvenslistor samt lagt till många ord för att knyta ihop dependenskedjorna, så att jag f.n. har en fil med drygt 6.000 komplexa "lemman" av följande utseende:

metallindustri ← [industri + metall]
plastindustri ← [industri + plast]
industriarbetare ← [arbetare + industri]
industri ← tillverka
tillverka ← göra
göra ← PRIM

Dvs de har alla strukturen A ← [B (+ C)].

Den typ av lemman som jag skapar skiljer sig från de traditionella inte bara i det avseendet att de är komplexa, alltså ett slags "superlemman", om man så vill, utan också därigenom att polysemi beaktas. Ett morfologiskt lexem klyvs i två semantiska, om de har helt eller partiellt olika dominanter, t.ex. *byte → byta, byte1 - fånga; bruka → vanlig, bruka1 → använda, bruka2 → odla*. I undantagsfall kan två lexem med identisk enda förälder klyvas, under förutsättning att "barnaskarorna" är klart åtskilda, t.ex. *växt* (→ *blad, blomma*) och *växt1* (→ *missväxt*), vilka båda återförs på *växa*.

Alla dependenskedjor avslutas i toppen med ett fiktivt lexem, *PRIM*. Jag har hittills definierat ca 50 lexem, som domineras enbart av *PRIM* (se Appendix). I och med att jag har arbetat igenom ett grundlexikon har till stor del lemmatiseringsprinciperna utkristalliserats. Eftersom även de mest frekventa orden, där ordbildningsrelationerna inte i så hög grad bestämmer lexemens betydelser, finns medtagna, kan man säga att den svåraste delen av arbetet är gjord.

Ur denna fil kan man - lämpligen automatiskt, alltså medelst ett dataprogram - generera ett ganska stort antal lexempar. Härvid tillkommer, förutom den direkt givna vertikala föräldrar-barn-relationen, två horisontella relationer, äkta-make-relation och syskonrelation, alltså:

a) A ← B; dvs A domineras av B (föräldrar-barn-relation)
b) B + C; dvs B kan sättas samman med C ("äkta-make-relation")
c) A # D; dvs A och D har (minst) en gemensam dominant ("syskonrelation")

Några exempel på lexempar som genereras ur de ovan anförda komplexa lemmana:

a) metallindustri ← industri, metallindustri ← metall;
   industri ← tillverka
b) industri + metall; arbetare + industri
d) metallindustri # plastindustri

Det genererade materialet kan listas på olika sätt, men ett överskådligt sätt är att ge samtliga ingående lexem i bokstavsordning och till varje lexem ge varje tänkbar direkt relationspartner, med relationen specificerad enligt nyssnämnda klassifikation. Den i sammanhanget centrala a-relationen, som definierar själva dependensen, har jag för listningsändamål hittills valt att klyva i två varianter, en med dominerat led på första plats, såsom i utgångslemmat (←), och en omvändning (→). Jag får därmed ordboksentryn av följande utseende:

industri ← tillverka
→ industriell, plastindustri, industriarbetare...
+ metall, plast, verkstad, arbetare...
# fabrik, producera...[← tillverka]

Det är emellertid klart att det behövs ytterligare specificeringar. Inom den vertikala relationen visar redan pilen dependensens riktning, men dessutom behöver anges om det är fråga om dominans från moder, fader eller ensamförälder. Inom äkta-make-relationen behöver anges riktningen i det fall då vi har att göra med huvudled och determinand (vi behöver t.ex. kunna skilja på *avfallsindustri* och *industriavfall*). Inom syskonrelationen, slutligen, behöver vi skilja på om lexemen har gemensam moder, gemensam fader, båda gemensamma, eller om det föreligger en mer avlägsen relation, som i paret *metallindustri # industriarbetare*, där dominanten är moder i ena fallet och fader i det andra (könsbyte!). Särskilt nära relaterade är fall med gemensam moder. Man bör observera att syskonrelationen, till skillnad från de övriga, är potentiellt mångställig, dvs att det här är mindre relevant att tala om lexempar. Vi får snarare serier av typen *kaffe, te, vin, öl...* (sammanhållna av hyperonymen *dryck*). Det visar sig för övrigt att just den intuitivt kända graden av samhörighet inom syskongrupperna utgör en god kontroll på att kopplingarna blivit riktiga gjorda vid lemmatiseringen.

Här ges ytterligare ett exempel, med dels flera icke-derivationella samband, dels mer specificerade relationer:

ved ←1 bränsle; ←2 träd
→1 björkved; →2 vedbod
+1 björk; +2 bod
#1 bensin, kol, olja [gemensam "mor"]; #2 bark, dunge, trädgren [gemensam "far"]; #3 al, asp, björk; bränsletank, bränslecell [gemensam "mor/far"]

Om man listar materialet på så sätt att varje lexem får utgöra ett eget entry, som ovan föreslogs, får man problem framför allt med redundans inom syskonrelationen. Om något lexem dominerar ett stort antal andra lexem (t.ex. *inte*, som är fader till alla lexem med en negativ komponent, såsom adjektiv av typen *ledsen, omöjlig*), ger detta faktum upphov till ett mycket stort antal symmetriska lexempar av typen *ledsen # omöjlig*. Här bör man alltså lägga in något slags

begränsning i programmet, t.ex. att man avbryter listningen efter den tionde partnern och hänvisar till det gemensamma dominerande lexemets entry.

Jag skall nu gå vidare i arbetet och enligt dessa principer lemmatisera en ordformsfil som genererats ur en textkorpus som består av artiklar ur tidskriften *Forskning och Framsteg* och som omfattar nära 100.000 löpord. Det är enligt min åsikt flera fördelar med att arbeta utifrån en korpus, snarare än utifrån ett traditionellt lexikon. Ett av de avgörande skälen är att man lätt kan inkorporera även encyklopedisk kunskap, något som kan vara värdefullt bl.a. vid informationssökning. Denna kunskap läggs huvudsakligen in via egennamnen, framför allt givetvis sådana egennamn som inherent har en specifik referens, t.ex. geografiska namn. Följande exempel visar vilka lemmatiseringsprinciper jag här har tillämpat: Olof ← förnamn, Fido ← namn + hund, Olof1 ← Palme, Palme ← politiker, Einstein ← fysiker, Malmö ← stad, Madrid ← huvudstad + Spanien, Vänern ← sjö, Ladoga ← sjö + Sovjetunionen etc. Exemplet *Olof Palme* visar att dependensrelationen även kan användas för att binda samman delar av fasta syntagmer, i de fall då man fortfarande vill ha de ingående komponenterna åtkomliga.

* * *

En annan typ av lexikon skall jag här skissera mycket kort; jag befinner mig ännu endast i början av materialinsamlingen. Detta lexikon bygger på hypotesen, alternativt postulatet, att varje lexem omger sig med ett kognitivt nätverk, där det självt upptar en nod. Uppgiften består i att inventera dess "nära nodgrannar", alltså de lexem som det har en direkt relation till, och att specificera denna relation i semantiska, kvalitativa termer. Detta förutsätter att vi har bestämt oss för vissa principer för att representera kunskap. Mitt förslag är att representera kunskap med naturliga lexem i kombination med en uppsättning logisk-semantiska valensramar. Jag arbetar sedan några år tillbaka med följande ramar:

A		KO		
B	KS	KO		
C		KO	KO'	
D	KS	KO	KO'	
E		KO	KO^-	KO^+
F	KS	KO	KO^-	KO^+
G		KO	KL	
H	KS	KO	KL	
I		KO	KL^-	KL^+
J	KS	KO	KL^-	KL^+

Systemet gör först och främst en distinktion mellan objekt (T) och situationer (P), härovan generaliserat till K ("kategori"). Som synes bestäms konfigurationerna av dels antalet aktanter, dels dessa aktanters semantiska roller. Denna klassifikation avspeglar grundläggande logiskt-semantiska egenskaper som terminativ, agentiv, lokalistisk m.m. Jag använder mig av tre grundläggande semantiska roller, S, O och L; i terminativa situationer, alltså sådana som inne-

bär egenskapsförändring, förvandling eller förflyttning klyvs dock O resp L i två faser, markerade med + resp -. Roller kan vidare sammanslås och redupliceras, men jag skall inte gå in på detta här.

Det nätverk som sålunda kan byggas upp runt varje lexem har en nära motsvarighet i de s.k. "lexikaliska funktioner" som har använts av Mel'čuk i hans språkmodell "meaning-text" och som har en framträdande plats i det lexikon - "Explanatory Combinatorial Dictionary" (1984) - som han och hans medarbetare har sammanställt. Exempel på dessa lexikaliska funktioner är: Magn(sjuk) = svårt; Oper1(inflytande) = utöva; $Caus_{loc}$(fartyg) = varv. Tanken var ursprungligen att man med hjälp av dessa funktioner skulle reducera antalet enheter på en mer generell, "djupare" representationsnivå, t.ex. ett interlingua för automatisk översättning.

Funktionerna är meningsfulla att använda endast när värdet på dem vid ett givet argument är åtminstone någorlunda bestämt, entydigt. Man kan därmed säga att de definierar den lexikaliska determinismen i ett givet språk, inbegripande dels icke förutsebar särspråklig information, dels kunskap om världen, eller i varje fall kunskap om hur det typiskt ser ut i världen, det som i olika sammanhang kallats "frames", "scripts", etc.

Att representera dessa funktioner genom hänvisning till olika positioner i ett kognitivt nätverk erbjuder, enligt min uppfattning, ett enklare och mer generellt system än det mel'čukianska. Gentemot det hierarkiskt uppbyggda system, som jag nyss har presenterat, har det den fördelen att man kan redovisa alla relationspartner till ett givet lexem; man behöver med andra ord inte välja mellan t.ex. 'näsa' och 'ficka' som dominant vid lexemet *näsduk* utan kan redovisa båda, med angivande av dels arten av semantisk relation, dels determinism och eventuellt relevans:

B/J:	$TO_b \leftarrow TL_j^-$	näsduk - näsa (B = använda, J = snyta)
H:	TO←TL	näsduk - ficka (H = förvara)
C:	TO←TO'	näsduk - duk; (C = "kind-of")

etc., där / anger att två situationer är involverade och pilen markerar determinismens riktning; i bokstavskombinationen TO, TL etc. betecknar den första kategori, den andra roll.

Om man i stället för ett ensamt lexem låter ett lexempar utgöra argument i funktionen ökar determinismen i nätverket avsevärt. Här kan vi också få in associationer av typen *elefant* : *grå*, *brandpost* : *röd* (se Charniak 1983, 442 ff), jfr även *häst* : *havre*, *kanin* : *morot*, alltså sådana för vilka man ofta använt en "attribute-value-struktur", se bl.a. Woods (1975, 50 ff). Dessa missar man ofta i den enklare hierarkiska strukturen. Å andra sidan har attribute-value-strukturen en begränsad tillämpning: vilket "attribut" skall man t.ex. ansätta vid relationen *damm* (= eng. *dust*) : *torr*? Endast relativt sällan är leden i ett givet lexempar förenade på mer än ett sätt, dvs så att det i en tredje nod tillåts mer än ett lexikaliskt värde, jfr:

B:	TS↔TO	bonde - åker (B = bruka)
J:	$TS \leftrightarrow TL^+$	bonde - åker (J = så, gödsla)

Många rollkonfigurationer tillåter olika kategoribeteckningar, jfr:

G:	TO→TL	blomma - vas
G:	PO←→TL	sova - säng
G:	PO←PL	svettas - arbeta

Syftet med båda typerna av ordböcker är att lämna ett bidrag till det nödvändiga arbetet med att inventera de otaliga intersubjektiva associationer, vilka utgör en förutsättning för språkförståelsen. Dessutom finns mer konkreta tillämpningar, såsom en semantiskt baserad informationssökning och automatisk textindexering, där en inventering av detta slag kan visa sig mycket användbar.

Referenser

Bring, S. C., *Svenskt ordförråd ordnat i begreppsklasser*, Uppsala 1930.

Charniak, E., Context Recognition in Language Comprehension, *Strategies for Natural Language Processing*, London etc. 1982, s. 435-454.

Kassabov, I., On the Problem of Defining the Core of the Vocabulary of the Bulgarian Language, *Linguistique Balkanique* 30:1 (1987), 51-55.

Mel'čuk, I. A. & Žolkovskij, A. K., *Tolkovo-kombinatornyj slovar' sovremennogo russkogo jazyka*, Wien 1984. (= Wiener Slavistischer Almanach, Sonderband 14).

Ralph, B., Primitiva verb i svenskan, *Förhandlingar vid sammankomst för att dryfta svenskans beskrivning* 11, Stockholm 1979, 163-174.

Roget's Thesaurus of English Words and Phrases, London 1962 (Longmans).

Wierzbicka, A., *Semantic Primitives*, Frankfurt/M 1972.

Woods, W. A., What's in a Link: Foundations for Semantic Networks, *Language, Thought, and Culture*, New York etc. 1975, s. 35-82.

Appendix: Lexem dominerade av PRIM

all	före	ljud	och	vad
annan	göra	ljus (subst)	om	var
använda	ha	med	på	vara
bara	hur	men	rak	varm
bra	hända	mot	röra	vem
den (art)	i	mycken	så	veta
fort	inte	måste	säga	vid
framme	just	namn	tal	vilja
färg	kant	natur	till	än
för	kunna	när	tänka	

LÆSNING AF MASKINLÆSBARE TEKSTER

Ole Norling-Christensen
Gyldendals Ordbøger
Postboks 11
DK-1001 København K

1. Indledning

Så længe maskinlæsbare tekster kun læses af de maskiner, de er beregnet for, giver det sjældent problemer. Vanskeligere bliver det, når man ønsker at genbruge teksterne i andre maskiner eller til formål, som ikke var forudset, da teksterne blev til.

Hensigten med denne artikel er, at

- give en oversigt over vanskelighedernes art,
- afgrænse de tilfælde, hvor de kan overvindes med rimeligt enkle midler,
- orientere om de seneste års standardisering på området, og
- foreslå, at vi, der arbejder med disse ting, prøver at enes om et fælles udvekslingsformat for tekster som kunne være af interesse for flere.

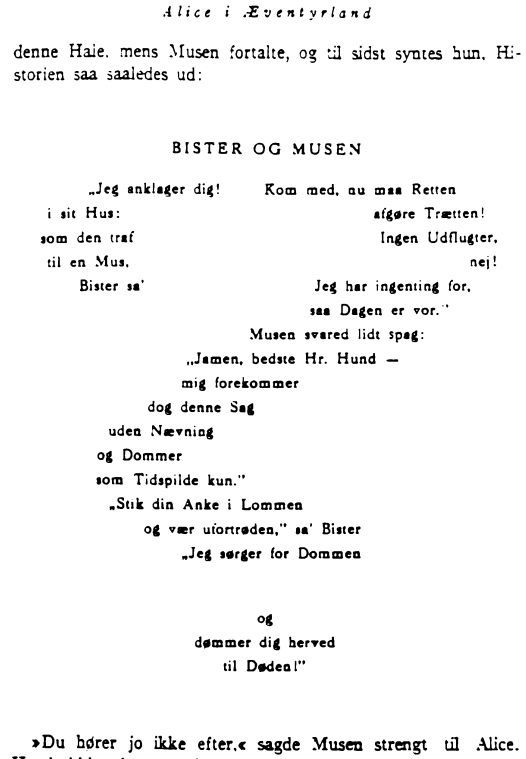
Problemstillingen er væsentlig både for nogle forlag og for datalingvister. Jeg har arbejdet en del med den i begge sammenhænge og kan derfor måske kaste en smule tværfagligt lys over den.

På forlaget møder vi problemet næsten daglig: en forfatter kommer med en diskette og siger: "Her er mit manuskript!" Skønt en fotosættemaskine kan modtage indtil 2000 tegn i sekundet, mens en tasteoperatør ikke kan præstere meget mere end 2 (Vail 1987), har vi - i hvert fald indtil for nylig - som regel måttet sige "Nej tak, vi vil hellere have en udskrift på papir". Det er galt nok; men værre er det, når det drejer sig om den slags tekster - typisk ordbøger og leksika - som over år eller tiår løbende skal revideres. Tidsintervallerne bliver her så lange, at de typografiske systemer i mellemtiden er blevet ændrede eller endda helt udskiftede.

For datalingvistikken måtte det være ideelt, at enhver tekst, som man ønskede at studere, blot kunne hentes på biblioteket i standardiseret form. En hindring herfor er, at det ikke altid er ganske klart, hvilke træk ved en trykt tekst den maskinlæsbare version skal afspejle. Herom mere nedenfor; lad mig foreløbig blot - uden yderligere kommentar - illustrere det med et citat (fig. 1, næste side).

2. Brug af maskinlæsbar tekst

Inden for den datamatstøttede leksikografi kan jeg umiddelbart pege på tre områder, hvor anvendelse af maskinlæsbare tekster kunne være et realistisk alternativ til egen (gen)indtastning: Datamatisk behandling af allerede eksisterende, trykte ordbøger; opbygning af korpuser m.m.; automatisk excerpering fra



Alice i Æventyrland

denne Hale, mens Musen fortalte, og til sidst syntes hun, Historien saa saaledes ud:

BISTER OG MUSEN

„Jeg anklager dig! Kom med, nu maa Retten
i sit Hus: afgøre Trætten!
som den traf Ingen Udflugter,
til en Mus, nej!
Bister sa' Jeg har ingenting for,
saa Dagen er vor."
Musen svared lidt spag:
„Jamen, bedste Hr. Hund —
mig forekommer
dog denne Sag
uden Nævning
og Dommer
som Tidspilde kun."
„Stik din Anke i Lommen
og vær uforfærden," sa' Bister
„Jeg sørger for Dommen
og
dømmer dig herved
til Døden!"

»Du hører jo ikke efter,« sagde Musen strengt til Alice. Hvad sidder du og tænker paa?«

»Undskyld...« sagde Alice meget høfligt. »Men var du ikke kommet til den femte Sneing?«

Fig. 1. Fra Alice i Æventyrland (Carroll 1946). Alice er optaget af at stirre på musens lange hale, mens musen fortæller denne "lange og bedrøvelige" historie om hunden Bister.

store tekstmængder, strømme af tekst, typisk avis- og telegrambureau-tekster.

Hvis allerede eksisterende, trykte, ordbøger skal lagres i databaser eller anden eksplicit struktureret form, til brug for revision og/eller alternativ præsentation, skal de ikke blot kunne læses af maskinerne; det læste skal også kunne strukturanalyseres, så de enkelte oplysningstyper og relationerne imellem dem klargøres. Hertil kræves i højere grad end for simple tekstundersøgelser en tolkning af oplysningernes grafiske fremtræden, eller rettere: en tolkning af de maskinlæsbare data som repræsenterer denne fremtræden.

Korpuser og konkordanser, samt automatisk eller blot maskinunderstøttet excerpering, kræver - ligesom enhver anden datalingvistisk undersøgelse - tekster i naturligt sprog, som kan læses af en datamat. Men indtil videre er de fleste korpuser fremstillet ved (gen)indtastning eller optisk læsning (OCR, Optical Character Reading) af trykt tekst - med efterfølgende korrekturlæsning - skønt mange af de pågældende tekster faktisk allerede fandtes i maskinlæsbar form.

Automatisk excerpering er behandlet af Robert Amsler (1986), som bl.a. indfører begrebet "NewsWire Lexicography". Fra en tekststrøm på flere hundrede tusinde ord pr. dag fra telegrambureauet Associated Press uddrager han, ved scanning efter ret enkle kriterier, leksikografisk interessante passager. Fx finder han implicitte definitioner af nye ord og udtryk ved at søge efter forekomster som "acronym for", "defined as", "usually called", "new name". Metoden foregribes af Pia Riber Petersen (1984:18-19) i hendes beskrivelse af, hvilke træk i et

belæg der tyder på, at et dansk ord er en leksikalsk nyhed; blandt de signaler, hun nævner, er citationstegn, samt ord som "såkaldt", "kaldes", "ordet", "dvs.". Lenders (1986) har brugt en tilsvarende fremgangsmåde til at finde centrale begreber hos Kant og afklare deres indhold. I det danske kulturministeriums regi undersøges det for tiden, om dagbladsartikler kan formidles til læsehandicappede via datatransmission / syntetisk tale. Hvis dette bliver en realitet, må også dansk leksikografi kunne drage nytte af denne tekststrøm.

Hovedtesen hos Amsler er, at maskinlæsbar tekst med fordel kan udforskes langt mere, end det sker i dag, med henblik på at skaffe leksikalske oplysninger. NewsWire Lexicography er kun ét eksempel blandt de mange muligheder han foreslår. Men desuden gennemgår han de vanskeligheder, som især data fra den grafiske industri, typisk fotosætterier, frembyder: Den typografiske kodnings eneste formål er at frembringe et bestemt visuelt mønster på papir, ikke at bevare informationsindholdet i en form, der kan bruges til andre formål. Også det modsatte gælder, anfører han: Der mangler ligeledes faciliteter til at omsætte databasers indhold til velformet trykt tekst; de eksisterende rapportgeneratorer er ikke tilstrækkelige.

Ifl. Amsler savnes der altså nye måder at repræsentere maskinlæsbar information på, metoder der både tjener det formål, som teksten oprindelig produceredes til, og nye formål, som kan være vidensbaserede edb-programmer henholdsvis grafiske produkter: "I have great interest in solving this problem, but no quick solutions to offer ...".

3. Tekstformidlingsprocessen

En del af meningen med denne artikel er at gøre opmærksom på, at sådanne nye repræsentationer faktisk er ved at blive fastlagt i form af en række internationale standarder. Men før jeg går nærmere ind på dem, vil det være nyttigt at analysere problemstillingen "læsning af maskinlæsbare tekster" lidt nøjere, prøve at fastslå problemernes art. Som udgangspunkt bruges et norsk forslag til niveaudeling af tekstformidlingsprocessen (fig 2, næste side); det er udarbejdet af Geir Andresen (1987) fra INGRAF, det norske institut for grafisk forskning.

Ideen er, at vejen fra forfatter over forlag eller redaktør til den færdige grafiske præsentation kan opdeles i mange små trin eller niveauer. På hvert trin træffes der - og kan der kun træffes - bestemte slags beslutninger: Underliggende niveauer skal ikke påvirke beslutninger som er truffet på niveauer højere oppe i skemaet, men understøtte disse. På den anden side må beslutninger truffet på ét niveau baseres på kendskabet til mulighederne på de underliggende niveauer, idet alt, som ligger over et niveau, skal kunne understøttes af det apparat, som er tilgængeligt på niveauet.

(1) På første (øverste) trin foreligger der "noget", der skal formidles som tekst, - tanker der endnu ikke er færdigt udformet i ord eller billeder.

Trinn ved tekstformidling

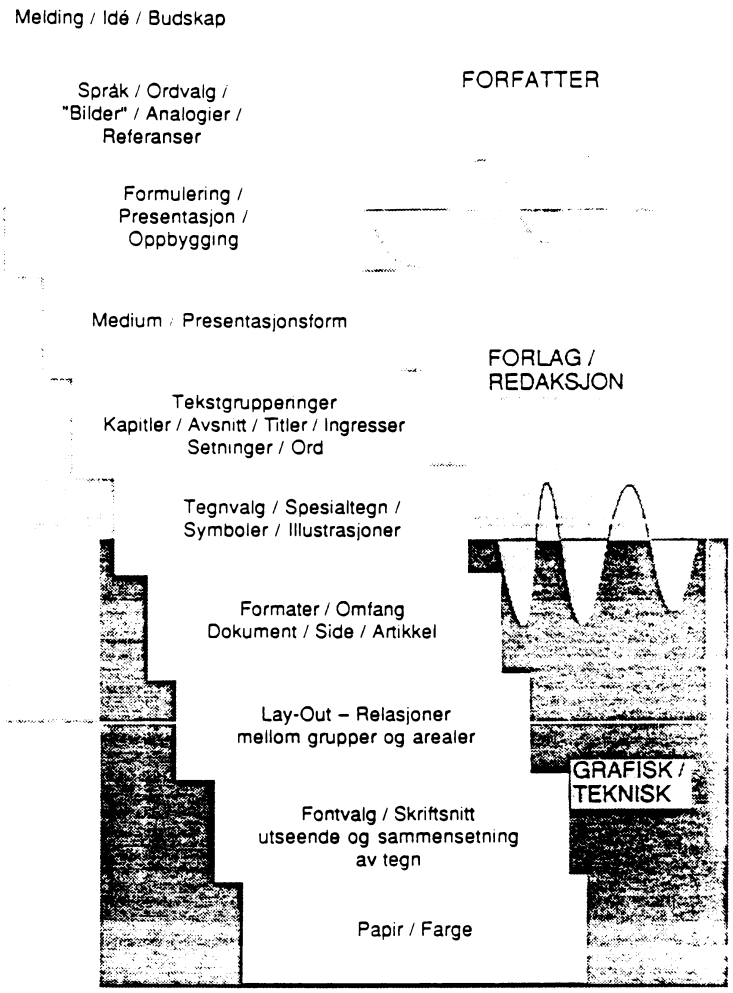


Fig. 2.
Efter Andresen (1987).

Underliggende nivåer skal ikke påvirke beslutninger som er gjort på overliggende nivåer, men underbygge disse. Overliggende beslutninger gjøres ut fra erfaringer fra underliggende nivåer.

(2) Dernæst sættes tankerne i ord; metaforer, henvisninger og sammenligninger udtænkes for bedst muligt at formidle budskabet.

(3) På tredje trin formuleres og disponeres teksten; den opdeles i indledning, forskellige afsnit og underafsnit, slutning.

(4) Valg af medie/præsentationsform er en vigtig beslutning, men svær at placere i skemaet. I den viste version er det anbragt som fjerde trin, i en anden version først som sjette. Her besluttes det, om teksten skal blive til en bog, en tidsskriftartikel, eller fx en avisnotits.

(5) Indholdet og dets rækkefølge er lagt fast. Tilbage står den detaljerede opdeling af teksten i overskrifter, brødtekst, lister, tabeller, fodnoter, billedtekster. Her træffes ikke længere beslutninger om indholdet, kun om formen. Men, i hvert fald i princippet, er der tale om generisk - artsmæssig - beskrivelse: Det markeres, at der er tale om forskellige slags

tekst; men der siges stadig intet om, hvilke skriftsnit og andre grafiske virkemidler der skal bruges til at udtrykke dem.

(6) Nu skal der tænkes på illustrationer, herunder specialtegn, lydskrifttegn, små "billeder i teksten", "ikoner", "sigler".

(7, 8) På trin 7 fastlægges de generelle typografiske regler for det pågældende medie, og på trin 8 indpasses den konkrete tekst heri.

(9, 10) Endelig handler de to nederste trin om selve sætningen og trykningen af teksten.

4. Fire slags maskinlæsbar tekst

Skemaet er - naturligvis - en abstraktion; i det virkelige liv vil hverken forfatter, forlag eller teknik opleve grænserne så skarpt. Afgrænsningen, rækkefølgen og antallet af forskellige trin kan diskuteres; som nævnt er Andresen selv i tvivl om placeringen af medie/præsentationsform. Den diskussion skal ikke tages her. Skemaet skal blot bruges til at klargøre, at der er forskellige slags maskinlæsbar tekst, og at forskellene bl.a. skyldes, at de hører til forskellige trin eller niveauer på tekstens vej fra tanke til tryk.

Omkring trin 2-3 er der tale om, hvad vi kunne kalde rå tekst, ord der er ordnet i sætninger, som måske igen er ordnet i afsnit, den slags tekst, som vil være nødvendig og tilstrækkelig for de fleste datalingvistiske undersøgelser, den slags tekst, som en traditionelt arbejdende ordbogsredaktør ville overføre udsnit af til sin excerptseddel.

Omkring trin 5 kommer den generiske mærkning ind, den der opdeler materialet i forskellige slags tekst (fx forord, kapitler, noter) uden at beskrive, hvordan forskellene skal vises i det færdige produkt. Også denne mærkning kan i visse tilfælde være nyttig for leksikografen eller datalingvisten.

Omkring trin 7-8 tilføjes layout mærkningen. Det kan i mange tilfælde gøres alene på basis af den generiske mærkning samt nogle generelle regler for det pågældende layout: nu skal det fx besluttes, om noten skal være en fodnote eller stå i et særligt noteafsnit, om en given fremhævelsestype skal vises med kursiv eller kapitæler. På dette niveau beriges teksten med koder for forskellige skriftsnit (kursiv, fed) og skriftstørrelser. Ofte vil der, især hvis der ud over teksten også er billeder, ikke kun blive taget hensyn til de forskellige tekststykkers art, men også æstetiske hensyn, fx til sidernes udseende. Under alle omstændigheder bliver den generiske mærkning hermed igen overflødig for så vidt angår det konkrete grafiske produkt. Derimod vil den bevare sin værdi for alle andre anvendelser af teksten: anden grafisk udformning, electronic publishing, datalingvistiske analyser. Ofte vil det være denne sidste slags maskinlæsbar tekst, et fotosætteri vil kunne bidrage med, og det kan være ganske svært at rekonstruere den rå tekst eller den generisk kodede tekst herfra.

Ved de to nederste og sidste trin er vi dybt inde i sættemaskinens og trykkemaskinens indre. Herfra vil vi næppe få brugbare

maskinlæsbare data; men vi vil få en trykt tekst, hvis indhold gerne skulle svare til det, vi fik ud af at maskinlæse tekster fra et af de tidligere niveauer. Dog kan det være et problem for traditionel filologisk anvendelse af de maskinlæsbare tekster, at ombrydning i linier (, spalter) og sider ofte først sker på disse sidste trin.

Alt efter, på hvilket tidspunkt af processen teksten gøres maskinlæsbar (eller gives fri til forskningsformål), kan vi altså skelne mellem rå tekst, generisk kodet tekst, typografisk kodet tekst og egentlige grafiske data (fx bit-map, en redegørelse for, hvor på siden der skal være sort, og hvor hvidt). Rå og generisk kodet tekst fås fx fra tekstbehandlingsanlæg; generisk og typografisk kodet tekst fra sætterier. De egentlige grafiske data hører til i og lige før sættemaskinen (jf. fig. 3).

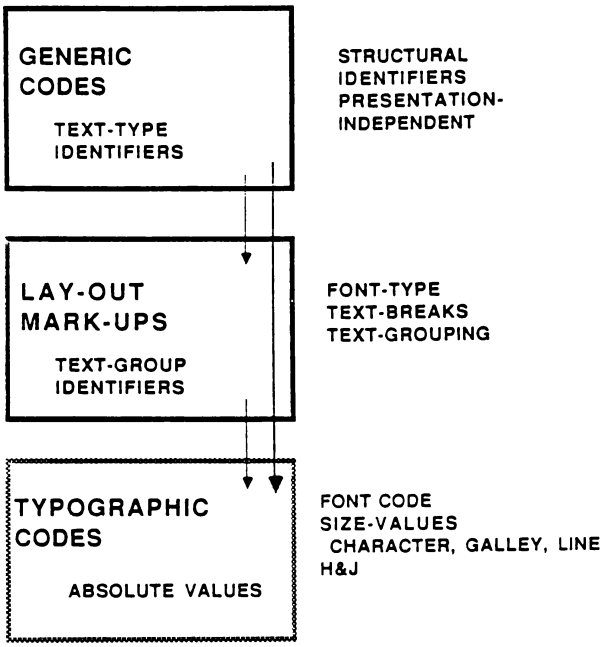


Fig. 3. De tre slags kodet maskinlæsbar tekst, som i artiklen kaldes generisk kodet, typografisk kodet, og egentlige grafiske data. Efter Andresen (1987).

Bortset måske fra de egentlige grafiske data vil den rå tekst i princippet stadig være til stede som koder, der repræsenterer enkelttegn (bogstaver, interpunktionstegn, osv.). Men jo længere vi kommer ned i skemaet, desto mere er den rå tekst gemt imellem alle mulige andre koder. Linier hørende til to forskellige spalter står nu måske sammen, og i værste fald foreligger teksten blot som en bit map.

Opgaven ved læsningen vil være at afdække den rå tekst samt så meget af den generiske kodning, som der er brug for til den påtænkte anvendelse; endvidere at kunne videregive resultatet i en form, som er entydig, og som er brugbar for anvenderen.

5. Genotype / fænotype

Ud fra synspunktet "læsning af maskinlæsbar tekst" kan et trin i Andresens model anskues således: Der foreligger en maskinlæsbar tekst fra det foregående trin; denne tekst undergår en transformation, hvis inddata er (1) uddata fra foregående trin, og (2) nogle yderligere regler, som hører til trinet. Som nævnt

ovenfor kan fx layout-mærkninger eventuelt foretages maskinelt (trin 7-8); i så fald vil de "yderligere regler" være eksplicit formulerede. Men reglerne kan også være en grafikers mere eller mindre intuitive viden. I begge tilfælde gælder det imidlertid, at trinets uddata er en entydig funktion af to variable, nemlig inddata og de nye regler. Derimod er uddata i almindelighed ikke således beskafne, at inddata entydigt kan rekonstrueres ud fra dem.

To eksempler, baseret på erfaringer fra ordbogsområdet, kan belyse dette:

(1) Om et skriftsnitmærke (fx ordinær-, kursiv-start) står før eller efter et blanktegn eller et punktum kan ikke ses i det færdige produkt; det er derfor ligegyldigt, og selv den bedste korrekturlæser vil ikke have haft anledning til at ændre derved: genotypen (den kodede repræsentation) er forskellig; men fænotypen (det synlige resultat) er det samme. Der kan også forekomme flere skriftsnitmærker efter hinanden, af hvilke kun det sidste har betydning for den færdige, trykte teksts udseende.

(2) Gennem flere trin i modellen kan det være underforstået, at startparentes med hensyn til skriftsnit følger den efterfølgende tekst, slutparentes den foregående. Dette giver normalt "pæne" resultater; men en i ordbogen hyppigt forekommende oplysningstype står i parentes, indledes med ordinær (opret) skrift, og afsluttes med kursiv (skrå skrift). I korrekturen konstateres, at dette giver et visuelt "grimt" resultat: kombinationen af ordinær startparentes med kursiv slutparentes giver ikke det rette indtryk af, at der er tale om én og samme parentes. Det besluttes derfor, at alle parenteser skal være ordinære. Dette indbygges i selve sættemaskinen, på trin 9. Man definerer blot, at en "kursiv" parentes har samme udseende som en "ordinær". De maskinlæsbare data er ikke blevet ændret; og igen resulterer forskellige genotyper i samme fænotype. Først når (dele af) det typografiske system udskiftes, viser problemet sig: det nye system "ved" ikke, at kursive parenteser skulle se ordinære ud.

Ved udarbejdelse af programmer, der skal tolke denne slags maskinlæsbar tekst, er det nødvendigt at tage hensyn til sådanne mulige flertydigheder.

Ordene genotype (anlægspræg) og fænotype (fremtoningspræg) er lånt fra genetikken. De er udmøntet af den danske plantefysiolog og genetiker W. L. Johansen (1857-1927). Analogien kunne trækkes så vidt, at eks. 1 ovenfor svarer til de tilfælde, hvor forskellige genotyper (pga. dominans) giver samme fænotype, mens eks. 2 er det tilfælde, hvor samme genotype (pga. forskelligt miljø) resulterer i forskellige fænotyper.

6. "Generisk" - hvad er det egentlig?

Med vilje har jeg i det foregående ikke givet nogen præcis definition af "generisk". Thi hvad der er artsmæssig og hvad der er typografisk mærkning af en tekst, afhænger i nogen grad af, fra hvilket synspunkt sagen anskues.

Fra ordbogsredaktørens synspunkt vil et større antal forskellige oplysningstyper inden trykningen skulle reduceres til fx tre skriftsnit: halvfed, kursiv, ordinær, i en vis udstrækning modificeret med forskellige slags parenteser og interpunktionstegn. Den generiske inddeling af oplysningerne reduceres altså til en - mindre detaljeret - typografisk. Men for typografen betyder en mærkning som "{k}" (jf fig. 8) ikke nødvendigvis "kursiv"; faktisk oversættes den i fotosætteriet til "brug format 2". Og "format 2" kan i sætteriet frit defineres. I praksis indebærer definitionen ikke blot kursiv skrift af bestemt snit og størrelse, men desuden at fx det franske orddelingsprogram skal benyttes. Ønskede man imidlertid en anden skrift end kursiv, kunne det lige så vel indkodes som en del af "format 2". Hvad ordbogsforfatteren må opfatte som en typografisk kodning, kan typografen altså med samme ret anse for en generisk.

7. Tekniske problemer

Maskinlæsbar tekst vil ofte blive leveret på magnetbånd eller diskette; men også direkte kommunikation fra én maskine til en anden forekommer. For alle tre medier gælder, at der findes flere forskellige standarder for, hvordan dataene lagres/transmitteres. Ikke mindst disketteområdet har været kaotisk: Der er 8", 5 1/4", 3 1/2" disketter. De kan være formatteret enkelt- eller dobbeltsidet og med forskellig tæthed, dvs. antal spor pr. tomme. Nogle er indspillet med konstant hastighed, andre med en hastighed, der varierer med læse/skrivehovedets afstand fra diskettens centrum. Filerne kan være organiseret under (MS/)DOS, CP/M eller et helt tredje operativsystem. Selv en 360 kB (MS/)DOS-diskette kan - hvis den er formatteret og/eller skrevet på et 1,2 MB drev - ikke altid læses på andre slags drev.

Dette er dog ikke det værste; som regel er det muligt at få en tekstfil overført til anden maskine, hvis man vil betale for det ("tekst" skal her forstås som den datatype der blot - set fra datamaten - er én vilkårligt lang, sekvens af tegn, uden anden datastruktur). Dermed er vanskelighederne imidlertid ikke forbi.

Alt efter hvilken konvention, der er brugt, vil tegnene tilhøre et alfabet med i alt 128 tegn (7-bit kode) eller 256 tegn (8-bit kode). Men betydningen af de enkelte tegn og tegnkombinationer afhænger bl.a. af, hvilket tekstbehandlingssystem der er anvendt. Selv ved tekster, der er skrevet med det samme tekstbehandlingsprogrammel, kan kodningen være forskellig, dels fordi visse tegn og koder kan defineres af brugeren, dels fordi ikke alle maskiner råder over samme tegnsæt; jf. 9.3 Tegnstandarder nedenfor.

Overførsel af tekstfiler fra et system til et andet er, som nævnt, mulig; bl.a. har UNI*C faciliteter hertil. Men dels er konverterings- eller kommunikationsudstyr dyrt, dels er en rent mekanisk overflytning kun sjældent tilstrækkelig. Den kan klare de fysiske forskelle på disketterne og flytningen fra et operativsystem til et andet. Generelle forskelle mellem de almindeligste tekstbehandlingssystemer kan naturligvis også klares med generelle konverteringsprogrammer; men så snart teksten rummer andre tegn end det engelske alfabet, kan det gå galt, med

mindre forlaget eller sætteriet for hvert enkelt manuskript lægger et stort arbejde i at lave individuelle konverteringstabeller. Rummer teksten fx skemaer og tabeller, eller kemiske og matematiske formler, er det næsten sikkert, at det går galt.

En undersøgelse (Møller 1987:8-9) viste da også, at kun 1/3 af de danske sætterier kunne modtage disketter, og at praktisk taget ingen af dem reklamerede for det. Indtil videre ser det altså ud til, at disketten kun er et realistisk alternativ til papirmanuskriptet for kunder med store mængder af tekster, der overholder en ensartet konvention.

Moderne tekstbehandlingsprogrammer får stadig flere faciliteter til at præsentere teksterne flot: fremhævelser, forskellige skrifter, tabulering, lige højremargen, forskellige spaltebredder, flerspaltet tekst, administration af fodnoter og registre. Desk-top Publishing er blevet et modeord. Men generelt kan det siges, at jo mere forfatteren har udnyttet sådanne grafiske faciliteter, desto vanskeligere er det at få et andet system til at læse og tolke teksten korrekt. Selv de firmaer, der lever af at sælge diskettekonverteringsudstyr, anbefaler derfor (Vail 1987), at man i stedet anvender generisk kodning. Denne kan aftales individuelt mellem sætteri og kunde. Men der findes også både typografisk orienterede standarder (INGRAF 1985; Cave 1986) og det helt generelle SGML, som omtales nedenfor.

Alt dette nævner jeg af to grunde. Dels fordi sprogligt materiale fra tekstbehandlingsanlæg også kan være af interesse for datamatstøttet leksikografi og anden datalingvistik, dels fordi problemerne med tekst fra avancerede tekstbehandlingssystemer ligner dem, man støder på, når man prøver at læse tekst fra fotosætterier, fx avis- og bogtekster. Hvert enkelt sætteri har sit eget kodesystem - også selv om apparaturet er det samme. Og selv det samme sætteri kan have ændret sine koder siden sidst; det giver visse vanskeligheder, når fx en ordbog skal revideres hvert tredje eller femte år. En fordel ved sætteridata frem for visse "avancerede tekstbehandlingsdata" er dog, at sætteriet normalt vil råde over et ikke-ombrudt (typografisk-generisk) arkivformat, som bevarer de typografiske fremhævelser, men alligevel er rimeligt tilgængeligt for genbrug.

8. Praktiske erfaringer

Med henblik på fotosætning af tekst fra ordbogsredigeringssystemet Compulexis (bl.a. ODSS 1987); overførsel af ordbogsdata fra forskellige typografiske systemer (jf. figg. 4-9) til et redaktørorienteret tekstbehandlingssystem og tilbage igen, samt strukturanalyse af dataene med henblik på databaselagring; fremstilling af en retrogradordbog på basis af RO (1986); m.fl. opgaver; har jeg udviklet en række analyse- og konverteringsprogrammer. De er skrevet i Pascal og kan stilles til rådighed for andre, som vil forsøge sig på området.

Min grundlæggende erfaring har været, at det altid er nødvendigt at gennemføre en total analyse af kodestrukturen i den pågældende tekst. Thi enten er en beskrivelse ikke til at få fat i, eller også viser det sig, at den er ukorrekt eller ufuldstændig.

Fremgangsmåden er i øvrigt følgende:

(1) Kig overfladisk på dataene. Det kan fx gøres med faciliteten VIEW i programmet QuickDos (Gazelle Systems, Provo, Utah, USA), eller ved at udskrive nogle sider under anvendelse af en rutine der omsætter ikke printbare (kontrol)tegn til deres talværdi. Programmer til "hex-dump" kan naturligvis også anvendes. Eksemplet i fig. 4 viste at den pågældende fil - som var kopieret til diskette fra magnetbånd - indledes med en "header" og afsluttes med en "trailer", som begge var irrelevante. Efter traileren lå der desuden nogle helt tilfældige datarester:

```
700000001645Cpiovol_Header(0)1 567075606 aJob_
Archive Mon Dec 21 10:00:06 1987(10) (10)-ost /dev
/rmt/0yy CpiovolVersion3.1.1.1(10) (0) (0) (0) (0) (0) (0)
(0) (0) (0) (0) (0) (0) (0) (0) (0) (0) (0) (0) (0) (0) (0) (0)
00123-----00A(10)0707070004040212241007750001460000
144000002110373041631534210000160000070644600123--
---00A(0) (1)P5♥A(1)T(1)P4♥A(1)P1♥ Rei▒.(2)"(1)
P4♥A. (1)P2♥fk Academy: America: Associate; (1)P1♥
(i biografannonce) (1)P2♥(omtr.) (1)P1♥betinget ti
```

Fig. 4. Begyndelsen af Engelsk-dansk Ordbog (EDO 1988). Dele af headeren er oversprunget.

Det første kig kan også afsløre, at det foreliggende analyseprogrammel ikke uden videre kan anvendes. Russisk-dansk Ordbog, som blev lavet hos RECKU (nu UNI*C, København), foreligger således i en slags hulkortformat, hvor to "hulkort" / linier tilsammen bestemmer én linies tegn:

```
 I na præp.m.akk. 1 om retningen: (hen, ind, ned, om, op, over, ud) på;
+  ll ÜÜÜÜÜÜÜÜÜÜÜÜ l ÜÜÜÜÜÜÜÜÜÜÜÜ
  til; mod; i; v'ewat' ü st'enu hænge op på væggen; v'yglânut' ü dr'uga
+              kkkkkkkk+kkkkkkk                    kkkkkkkkkkk+kkkkkkk
  kaste et blik på sin ven; v'yjti ü 'ulicu gå udenfor; gå ud på gaden;
+                          kkkkkkk+kkkkkkk
  dor'oga ü Moskv'u vejen mod Moskva; 'exat' na S'ever rejse nordpå; leh'
+ kkkkkkkk+kkkkkkkk                  kkkkkkkkkkkkkkk                kkkk
  na div'an lægge sig (hen, ned etc.) på sofaen; perevest'i ü d'atskij
+ kkkkkkkkk                                      kkkkkkkkkk+kkkkkkkkk
```

Fig. 5. Udsnit af Russisk-dansk Ordbog (RDO 1985). To linier, den første indledt med blanktegn, den næste med "+" definerer tilsammen èn linies tegn. I "plus-linien" betegner "l" fed og "k" kursiv kyrillisk skrift; ü markerer kursiv, blanktegn ordinær latinsk skrift; "+" under "ü" betegner tilden, der gentager opslagsordet.

(2) Næste trin er at køre programmet TGNTAL, der udskriver første forekomst af hvert af de (højst) 256 tegn. Desuden optæller det antallet af forekomster af hvert enkelt tegn. Programmet giver mulighed for at overspringe indtil 32.767 tegn i starten. Herved kan fx den irrelevante "header" overspringes. Det er også muligt at overspringe tegnene A..Z, a..z, hvis det første kig har vist, at de blot repræsenterer sig selv, dvs. følger ASCII konventionen.

(3) Resultatet af denne første undersøgelse gennemgås nøje, idet de udskrevne tekstprøver sammenholdes med den trykte tekst. Ét resultat har hver gang vist sig: antallet af højreparenteser i ordbogsdataene er en smule lavere end antallet af venstreparenteser; nogle parenteser er altså ikke afsluttede, hvilket sidenhen kan give vanskeligheder, når parenteser skal bruges som kriterium for forekomsten af bestemte oplysningstyper (jf. fig. 6, næste side).

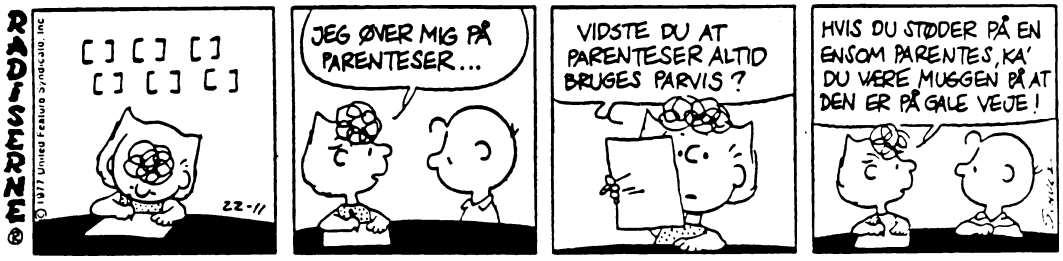


Fig. 6. Fra "Radiserne"; citeret efter dagbladet Politiken.

Gennemgangen afslører, hvordan hvert enkelt af de 128 eller 256 tegn er brugt, herunder fx hvordan æ, ø, å er repræsenteret. Desuden vil den vise, at visse tegn bruges til at indlede og/eller afslutte flertegnssymboler for grafiske tegn og andet, som ikke repræsenteres af enkelttegn:

```
93 ]    0.00%
        lse, sikkerhed (i optr{den), aplomb. (2)" (1)P4♥apoc
        alypse (1)P1♥R(17)fe(17)sjp(17)fdk(17)felips▒ (1)P
        2♥sb (1)P1♥)benbaring; (1)P2♥the A. (1)P1♥Johannes
        ' ]benbaring. (2)" (1)P4♥apocarp (1)P1♥R(17)sj{p
        (17)feka:p▒ (1)P2♥sb (bot) (1)P1♥ flerfoldsfrugt. (2)
        " (1)P4♥apocope (1)P1♥R(17)fe(17)sjp(17)fdk(17)fepi
        ▒ (1)P2♥sb (gram) (1)P
```

Fig. 7. For hvert konstateret tegn udskriver programmet ca. 127 tegn før og efter den første forekomst, som understreges. Desuden udskrives en tabel med det absolutte antal forekomster af hvert tegn. Her vises første forekomst af "Å" i (EDO 1988). Ved sammenligning med den trykte tekst konstateres det bl.a., at kontroltegnene <1>, <2> og <17> indleder flertegnssymboler.

(4) De næste to programmer bygger på den erfaring, at en "kode", dvs. et flertegnssymbol, ofte består enten af "kodestart" + et indhold af vilkårlig længde + "kodeslut" eller af et kodemærke + et fast antal tegn, som da hører med til koden.

Programmet STJRNTAL blev oprindelig lavet til at finde og optælle Gyldendals "stjernekoder", som både indledtes og afsluttedes med en "stjerne" (asterisk, ASCII-tegn nr. 42), heraf navnet; stjernerne er sidenhen afløst af { og } :

```
28e}    4.53%
        f2} {o}gå videre,; forlænge: {k}{f23} un mur; {f23
        } {26a}, de(13)(10){o}vedblive at; {f23} {k}q. dan
        s le m{2e}me emploi {o}lade {28e}n vedblive i(13)
        (10)samme virksomhed. {h}2. {o}vedvare, blive ved,
         fortsætte: {k}la pluie {f23}e;(13)(10){o}forlænge
        s, strække sig. {h}-it{2
```

Fig. 8. I Fransk-dansk Ordbog (FDO 1980) er venstre- og højre-"tuborg" tydeligt nok kodeindleder og -afslutter. Her vises første forekomst af koden for "è". Kontroltegnssekvensen <13><10> markerer postgrænser i filen ("ny linie"); de er irrelevante for ordbogsteksten og skal blot konverteres til blanktegn (ordmellemrum).

Programmet KODETAL kan i samme gennemløb af teksten behandle flere forskellige kodemærker, hvert med et individuelt defineret antal efterfølgende tegn; for nylig er det blevet udvidet, så det også kan "kigge bagud", idet en gruppe tekster viste sig at repræsentere accentbogstaver ved bogstavet efterfulgt af et symbol for accenten. Også dette program tæller antal forekomster af hver kode og udskriver den først fundne med kontekst:

<17>mg 4.95%
(1)P2Vfit able-bodied (seaman); (am (1)P1Vform for) (1)P2VB. A. (Bachelor of Arts). (1)P1V(2)" (1)P4Va back (1)P1VR(17)fe(17)sjb(k (1)P2Vadv: taken (17) mg (1)P1Vforbliffet. (2)" (1)P4Vabacus (1)P1VR(17) sj(b(17)fek(17)fes (1)P2Vsb (pl -es el. abaci (1) P1VR(17)sj(b(17)fesai) kugleramme, regnebr(t; (1) P2V(arkit)

Fig. 9. I (EDO 1988) repræsenterer kontroltegnet <17> efterfulgt af netop to andre tegn forskellige specialtegn, bl.a. fonetiske tegn. "<17>mg" er således tilden, der gentager opslagsordet.

(5) Konvertering af teksten til den ønskede form. Konverteringsprogrammet bør - for en sikkerheds skyld - udformes således, at det melder fejl og udskriver kontekst, hvis det møder tegn eller koder, der ikke udtrykkelig er defineret som tilladte.

9. Standarder

Et af problemerne ved at arbejde med maskinlæsbar tekst fra mange forskellige kilder er, som nævnt, at hvert system, hver leverandør, har sine egne koder. Yderligere kan man ikke regne med, at kodesystemet er fuldt dokumenteret.

Der er imidlertid håb om, at dette efterhånden vil ændre sig. Den internationale standardiseringsorganisation, ISO, har i de seneste år arbejdet intenst på at fastlægge standarder på området, og standardiseringsarbejdet støttes af både EF, USA og de nordiske lande. Faktisk har EF-landene - og dermed også Danmark - forpligtet sig til fra 1988 at stille krav om at standarderne overholdes ved alle offentlige indkøb af informationsteknologi og datakommunikation (Vejl. 1987).

9.1 Dokumentarkitektur

Blandt de mest ambitiøse af disse standarder er ISO/DIS 8613 (1987) Office Document Architecture / Interchange Format (ODA/ODIF). Den er endnu ikke vedtaget, men foreligger som udkast ("DIS" = Draft International Standard). Standardens formål er at muliggøre udveksling af kontordokumenter (rapporter, fakturaer, breve, notater osv.) ved hjælp af datakommunikation eller ved forsendelse af lagermedier (magnetbånd, disketter, etc.).

Skønt standarden specielt skal dække kontordokumenter, er den så generel, at den må kunne dække næsten alt andet også. Prisen for denne generalitet er, at standarden bliver så abstrakt og så omfattende, at næppe andre end specialister magter at sætte sig ind i den. Det er ikke en standard for kontorassistenter,

leksikografer eller datalingvister, men én som måske bliver indbygget i fremtidige tekstbehandlings-, informations- og kommunikationssystemer.

Ved "dokumentarkitektur" forstår ODA/ODIF "det sæt af regler, der angiver et udvekslet dokuments struktur"; og hovedideen er, at et dokument kan beskrives ved hjælp af to strukturer: en logisk struktur og en layout struktur. Den logiske struktur opdeler indholdet af et dokument i stadig mindre dele på grundlag af den måde, mennesker (logisk) opfatter indholdet på, dvs. i kapitler, afsnit, underafsnit, paragraffer og billeder. Layoutstrukturen opdeler indholdet i samlinger af sider, enkeltsider, arealer på siderne (fx spalter), linier. De to strukturer repræsenterer forskellige (og i princippet indbyrdes uafhængige), men komplementære syn på dokumentets indhold. Hver af dem beskrives som et hierarki af objekter; disse repræsenteres af såkaldte attributter, som definerer objekternes egenskaber og deres indbyrdes sammenhæng.

I Andresens skema (fig 2) indføjes den logiske struktur omkring trin 3-5, layoutstrukturen på trin 5-8.

Bortset fra filologers ønske om at kunne referere til en bestemt linie på en bestemt side i en trykt tekst, er layoutstrukturen mindre interessant i nærværende sammenhæng. Væsentligt er det blot, at den er adskilt fra den logiske struktur.

9.2 SGML, en standard for generisk mærkning

Den logiske struktur kan behandles i henhold til en anden standard, ISO 8879 (1986), Standard Generalized Markup Language (SGML), som er et system til mærkning af tekstdele efter deres art (den flere gange nævnte generiske mærkning). Selve standarden er tung læsning; men den ledsages af en række tillæg (Annex A - I), som rummer beskrivelser, der er mere tilgængelige for ikke-dataloger. Joan Smith (1986, 1987) har givet flere enkle introduktioner til emnet; som en god indføring kan desuden anbefales FORMEX (1985). Descriptive Tools (1987:139-44) giver en gennemgang, som især sigter mod leksikografiske anvendelser.

SGML er en generel formalisme, der beskriver et dokument som en hierarkisk struktur, et træ. Den kan virkeliggøres med vilkårlige symboler, af hvilke enkelte må reserveres til mærkningsformål. Syntaksen bygger på teorien for deterministiske finitte automater (tilstandsmaskiner) og muliggør derfor konstruktion af rimeligt enkle programmer, parsere, som er i stand til at behandle de strukturerede dokumenter. I gennemgangen nedenfor følges den SGML-konvention (FORMEX, 1985) som benyttes af EF's Kontor for Officielle Publikationer; de reserverede tegn er her "<" ("mindre end") og "&" ("og"), der benyttes på følgende måde:

En given kategori med navnet "xxx" indledes med koden "<xxx>" og afsluttes med "</xxx>". Mellem disse koder kan underordnede kategorier i vilkårligt mange niveauer indledes og afsluttes.

Længere udtryk, som hyppigt skal gentages, samt tegn der ikke er til rådighed i det anvendte alfabet, kan "stenografisk" noteres som "&yy;" - altså et navn, "yy" indledt af "&" og afsluttet af ";". Fx kunne "&SGML;" repræsentere "Standard Gene-

ralized Markup Language", og "&ag;" kunne repræsentere lille a med accent grave. Får man brug for de reserverede tegn, her "mindre end" og "og" tegnene, må de da symboliseres ved fx "&me;" og "&og;".

```
<entry>
  <hwsec>
    <hwlem id=1234567>Map</hwlem>
    <pron>(mæp)</pron>, <pos lit='sb.' hom=1>
    <vfl id=1234567>Also <vd>6–7 <vf>mappe</vf>, <vd>6–8
      <vf>mapp</vf>.
    <etym id=1234567>ad. L. <cf>mappa</cf>, in class.L. 'table-cloth,
      napkin', but in med.L. used <lab>transf.</lab> in the combination
      <cf>mappa mundi</cf> (see <xlem>Mappemonde</xlem>).
      <enote>Cf. the synonymous OF. <cf>mappe</cf> (rare: also in
        Rousseau <I>c<R>1770), Sp. <cf>mapa</cf>, Pg.
        <cf>mappa</cf>, G. <cf>mappe</cf> (obs.: the mod. sense
        'portfolio' is not directly connected).</etym>
  <signif s4=4>
    <sen4 par=1 lit='1.' s6=4>
    <sengp>
      <sen6>
        <stxt id=1234567>A representation on the earth's surface or a
          part of it, its physical and political features, etc., or of the
          heavens, delineated on a flat surface of paper or other
          material, each point in the drawing corresponding to a
          geographical or celestial position according to a definite scale
          or projection.
          <snote>A hydrographical map is now more usually called
            a <cf>chart</cf> (formerly † <cf>card</cf>).
      <qbank id=1234567>
        <quot id=1234567>
          <qdat>1527
          <srce><auth>R. Thorne</auth> in Hakluyt
            <wk>Voy.</wk> (1589) 257
          <qtxt>To make a bigger and a better mappe.
        <quot id=1234567>
          <qdat>1589
          <srce><auth>G. Harvey</auth> <wk>Pierce's
            Super.</wk> Wks. (Grosart) II. 130
          <qtxt>The great Mapp of Mercator.
        <quot id=1234567>
          <qdat>1601
          <srce><auth>Shaks.</auth> <wk>Twel. N.</wk>
            <SC>iii. <R>ii. 84
          <qtxt>He does smile his face into more lynes, then is in the new
            Mappe, with the augmentation of the Indies.
        <quot id=1234567>
          <qdat>1625
          <srce><auth>N. Carpenter</auth> <wk>Geog. Del.</wk>
            <SC>i. <R>vii. (1635) 166
          <qtxt>A Geographicall Mappe is a plaine Table, wherein the
            Lineaments of the Terrestriall Spheare are expressed.
        <quot id=1234567>
          <qdat>1760
          <srce><auth>Johnson</auth> <wk>Idler</wk> No. 97 ¶5
          <qtxt>A rivulet not marked in the maps.
        <quot id=1234567>
          <qdat>1867
```

Fig. 10. SGML-mærket tekst (korrektur) fra the New Oxford English Dictionary; den hierarkiske struktur er tydeliggjort ved indrykninger. Efter Benbow (1986).

Betydningen af kategorinavne og "stenogrammer" fastlægges i en dokumenttypedefinition (DTD), som nøje gør rede for kategorierne og de hierarkier, de kan indgå i, samt for de benyttede tegnsæt.

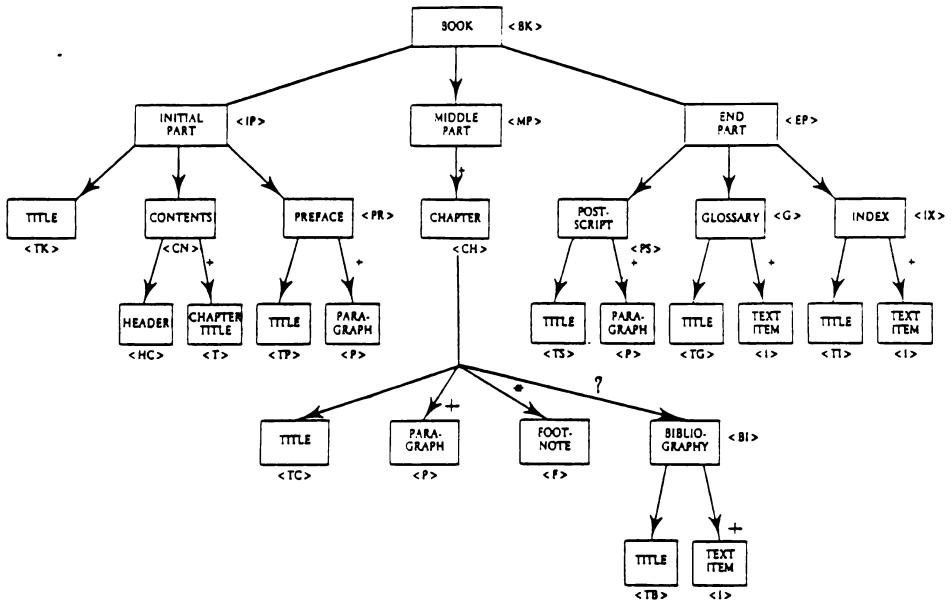


Figure 7

```
< !ELEMENT --          MIN  CONTENT --
  1 BK                 0 0  (IP, MP, EP)
  2 IP                 0 0  (TK, CN, PR)
  3 TK                 - 0  (#CDATA)
  4 CN                 0 0  (HC, T+)
  5 (HC|T)             - 0  (#CDATA)
  6 PR                 0 0  (TP, P+)
  7 (TP|P)             - 0  (#CDATA)
  8 MP                 0 0  (CH+)
  9 CH                 0 0  (TC, P+, F*, BI?)
 10 (TC|F)             - 0  (#CDATA)
 11 BI                 0 0  (TB, I+)
 12 (TB|I)             - 0  (#CDATA)
 13 EP                 0 0  (PS, G, IX)
 14 PS                 0 0  (TS, P+)
 15 TS                 - 0  (#CDATA)
 16 G                  0 0  (TG, I+)
 17 TG                 - 0  (#CDATA)
 18 IX                 0 0  (TI, I+)
 19 TI                 - 0  (#CDATA)
>
```

Figure 8

Fig. 11. En ret generel beskrivelse af dokumentstrukturen for en (fag)bog. Såvel i træet (øverst) som i dokumenttypedefinitionen (DTD) betegner "?" at det pågældende element forekommer ingen eller èn gang, "*" at det forekommer ingen, èn eller flere gange, og "+" at det forekommer èn eller flere gange; umærkede elementer forekommer netop èn gang. I DTD'en genskrives elementerne i venstre kolonne som vist i højre kolonne; "#CDATA" er en terminal kategori, nemlig tekst som ikke er yderligere opdelt, men kan rumme ethvert tegn fra et andetsteds defineret alfabet. Efter FORMEX (1985).

I dokumenttypedefinitionen kan det også fastlægges, at ikke alle kategorier behøver mærkning, idet bl.a. indledning af en sideordnet kategori, eller afslutning af en overordnet, vil være tilstrækkelig mærkning; kolonnen med "-" og "O" (= "omit") i DTD (fig. 11) angiver, i hvilken udstrækning dette skal være tilladt. Yderligere forenklinger kan opnås ved at definere forkortede mærkninger for hyppigt forekommende kategorinavne. Også de koder (kontroltegn m.fl.) som i tekstbehandlingsfiler markerer fx ny linie, nyt afsnit, tabulering og forskellige fremhævelser, kan defineres som forkortede SGML-mærkninger. Endvidere kan tekststrenge defineres som værende på én gang data og mærkninger. Med fuld udnyttelse af disse muligheder vil man næsten helt kunne undgå synlig mark-up selv i så kompliceret tekst som en ordbogsartikel (Erlandsen & Norling-Christensen 1988).

SGML-parseren er et program, som på basis af dokumenttypedefinitionen kontrollerer, at en given tekst er i overensstemmelse med den definerede struktur; desuden fuldstændiggør den mærkningen af tekster med reduceret / forenklet mærkning.

Mens jeg kan være i tvivl om, hvornår og i hvor høj grad ODA / ODIF slår igennem (den er som sagt meget ambitiøs), er det næsten sikkert, at SGML vil vinde frem. EF bruger det (FORMEX 1985); det er på vej ind i førende producenters tekstbehandlingssystemer; der er defineret dokumenttyper med bred anvendelighed, fx til sædvanlige videnskabelige afhandlinger (Smith 1987); og standarden er desuden taget i brug af store internationale forlag, især til videnskabelige tidsskrifter, som enten i deres helhed eller blot med abstracts og bibliografiske oplysninger skal indlægges i informationsbaser. Mærkningen betyder, at oplysninger om fx forfatternavn og -adresse, abstract, noter, bibliografi (og herunder de enkelte indførsler med forfatter, titel etc. identificeret), register m.v. kan genbruges i informationsbaser, selektive og generelle kataloger osv. Også til ordbogsarbejde er standarden taget i brug, først og fremmest af The New Oxford English Dictionary (Benbow 1986; Cowlishaw 1987; Descriptive Tools 1987:144-6); men flere SGML-baserede ordbogssystemer er på vej (Erlandsen & Norling-Christensen 1988). Gevinsten ved at anvende et system af denne art vil være, at ordbogsdataene kan bruges og genbruges, dels i forskellige medier, idet mange forskellige præsentationsformer kan afledes af den samme, medieuafhængige og veldefinerede, strukturbeskrivelse, dels i nye produkter som kombinerer data fra forskellige kilder.

9.3 Tegnstandarder

Hvorledes de 128 (7-bit kode) eller 256 (8-bit kode) forskellige bitkombinationer skal tolkes, afhænger af anvendelsen. De kan tolkes som heltal noteret i det binære talsystem (00000000 = 0; 01111111 = 127; 11111111 = 255), eller som bogstaver og andre grafiske tegn. Hvis en grafisk repræsentation ikke findes (visse kontroltegn), ikke er tilgængelig på en given printer eller skærm, eller det af andre grunde er ønskeligt entydigt at identificere en bitkombination, kan heltallet bruges, fx angivet som "<tal i titalssystemet>". Denne konvention er brugt i figg. 4, 7, 8 og 9, samt i det følgende.

ASCII (American Standard Code for Information Interchange) er den klassiske 7-bit standard; på et enkelt tegn nær ("$" =

<36>) er den identisk med den internationale referenceversion (fig. 12) af DS/ISO 646 (1974); denne standard er karakteristisk ved, at en række tegn ikke er internationalt definerede, idet det overlades til de nationale standardiseringsorganisationer at definere dem. I Danmark bruges de åbne pladser til æ, ø og å, i Tyskland til ä, ö, ü, ß , i Frankrig til ç og de øvrige accenterede bogstaver, osv. En sådan standard er anvendelig, så længe man holder sig til ét sprog, men klart uhensigtsmæssig, hvor flere sprog blandes.

#	Number sign	2/3
¤	Currency sign	2/4
@	Commercial at	4/0
[	Left square bracket	5/11
\	Reverse solidus	5/12
]	Right square bracket	5/13
{	Left curly bracket	7/11
|	Vertical line	7/12
}	Right curly bracket	7/13

Tegn (*Character*)	Position (Se ISO 646)	
	Kolonne (*Column*)	Række (*Row*)
Æ	5	11
Ø	5	12
Å	5	13
æ	7	11
ø	7	12
å	7	13

b7	b6	b5	b4	row \ column	0	1	2	3	4	5	6	7
0	0	0	0	0	NUL	TC (DLE)	SP	0	@	P	`	p
0	0	0	1	1	TC (SOH)	DC	!	1	A	Q	a	q
0	0	1	0	2	TC (STX)	DC	"	2	B	R	b	r
0	0	1	1	3	TC (ETX)	DC	£ (#)	3	C	S	c	s
0	1	0	0	4	TC (EOT)	DC	$ (¤)	4	D	T	d	t
0	1	0	1	5	TC (ENQ)	TC (NAK)	%	5	E	U	e	u
0	1	1	0	6	TC (ACK)	TC (SYN)	&	6	F	V	f	v
0	1	1	1	7	BEL	TC (ETB)	'	7	G	W	g	w
1	0	0	0	8	FE (BS)	CAN	(	8	H	X	h	x
1	0	0	1	9	FE (HT)	EM	)	9	I	Y	i	y
1	0	1	0	10	FE (LF)	SUB	*	:	J	Z	j	z
1	0	1	1	11	FE (VT)	ESC	+	;	K	[	k	{
1	1	0	0	12	FE (FF)	IS (FS)	,	<	L	\	l	|
1	1	0	1	13	FE (CR)	IS (GS)	-	=	M	]	m	}
1	1	1	0	14	SO	IS (RS)	.	>	N	^	n	¯
1	1	1	1	15	SI	IS (US)	/	?	O	_	o	DEL

Fig. 12. DS/ISO 646 (1974). Tegnene i kolonne 0 og 1 er de såkaldte kontroltegn; af de øvrige kan visse antage forskellige, sprogafhængige værdier. T.v. vises nederst de specielt danske værdier (DS 2089, 1974), øverst værdierne i standardens internationale referenceversion.

Dette tegnsæt kan udvides på to måder. For det første bestemmes det, at komma <44>, anførselstegn <34> m.fl. tegn, kombineret med <08> (BACKSPACE) repræsenterer cedille, trema, m.fl. diakritiske tegn: ligesom på en skrivemaskine "slår man to tegn oven i hinanden". Men desuden fastlægger en særlig standard, ISO 2022 (1986), hvorledes man kan bruge kontroltegnene <14> "Shift out", <15> "Shift in" og <27> "Escape" til at springe

til andre alfabet-definitioner, af hvilke der efterhånden findes adskillige.

Antallet af tilgængelige tegn er blevet udvidet gennem databehandlingens korte historie. Endnu for 10 år siden var 6-bit kode almindelig (64 tegn, kun versaler); i dag kan det meste udstyr håndtere 8-bitkoder og dermed op til 256 tegn ad gangen. Fælles for de ISO-standardiserede kodetabeller er det imidlertid, at grundstrukturen i ISO 646 bevares: hver tabel består af 32 kontroltegn og indtil 96 grafiske tegn. 8-bit standarderne kombinerer så blot to 7-bit tabeller, der lægges ved siden af hinanden. Herved bliver ikke blot kolonnerne 0 og 1, men også 8 og 9 reserveret til kontroltegn. Et eksempel herpå (fig. 13) er DS/ISO 8859-1 (1987), "Latinsk alfabet Nr. 1", som dækker de fleste nordiske og vesteuropæiske sprogs behov. ISO 8879 serien

b4 b3 b2 b1	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
0 0 0 0	00			SP	0	@	P	`	p			NBSP	°	À	Ð	à	ð
0 0 0 1	01			!	1	A	Q	a	q			¡	±	Á	Ñ	á	ñ
0 0 1 0	02			"	2	B	R	b	r			¢	²	Â	Ò	â	ò
0 0 1 1	03			#	3	C	S	c	s			£	³	Ã	Ó	ã	ó
0 1 0 0	04			$	4	D	T	d	t			¤	´	Ä	Ô	ä	ô
0 1 0 1	05			%	5	E	U	e	u			¥	µ	Å	Õ	å	õ
0 1 1 0	06			&	6	F	V	f	v			¦	¶	Æ	Ö	æ	ö
0 1 1 1	07			'	7	G	W	g	w			§	·	Ç	×	ç	÷
1 0 0 0	08			(	8	H	X	h	x			¨	¸	È	Ø	è	ø
1 0 0 1	09			)	9	I	Y	i	y			©	¹	É	Ù	é	ù
1 0 1 0	10			*	:	J	Z	j	z			ª	º	Ê	Ú	ê	ú
1 0 1 1	11			+	;	K	[	k	{			«	»	Ë	Û	ë	û
1 1 0 0	12			,	<	L	\	l	|			¬	¼	Ì	Ü	ì	ü
1 1 0 1	13			-	=	M	]	m	}			SHY	½	Í	Ý	í	ý
1 1 1 0	14			.	>	N	^	n	~			®	¾	Î	Þ	î	þ
1 1 1 1	15			/	?	O	_	o				¯	¿	Ï	ß	ï	ÿ

Fig. 13. DS/ISO 8859-1, Latinsk alfabet nr. 1. I denne som i den foregående figur findes et felts decimalværdi ved at gange kolonnens nummer med 16 og lægge liniens nummer til. "NBSP" (non-breaking space) er et blanktegn der ikke skal opfattes som ordmellemrum; "SHY" (soft hyphen) kan markere et potentielt orddelingspunkt. Venstre halvdel er ASCII alfabetet, højre halvdel et udvalg af andre latinske bogstaver m.m.

betragter "kombinerede bogstaver", dvs. ligaturer (fx dansk æ) og bogstaver med diakritiske tegn (fx ü), som enkelttegn. For at få plads til alle har en geografisk-sproglig opdeling i fire delvis ovelappende latinske alfabeter været nødvendig. Foruden nr. 1 rummer også nr. 4 (ISO/DIS 8859/4) de særlige danske tegn. Andre dele af denne serie og af en parallel serie, ISO 6937 (som bruger flertegnskombinationer for "kombinerede bogstaver"), omfatter desuden arabisk, græsk, hebraisk og kyrillisk. En del af de nævnte standarder er endnu ikke færdigbehandlede, men foreligger som forslag.

Standarder kan fastlægges på flere måder: af nationale eller internationale standardiseringsorganisationer, eller som "de facto industristandarder", hvor én tilstrækkelig indflydelsesrig producent (på dette område ofte IBM) sætter normen. De ovenfor omtalte, og specielt 8859-serien, understøttes - og er delvis udarbejdet - af ECMA (European Computer Manufactures Association); IBM's alfabeter afviger stærkt herfra. På større IBM-anlæg bruges EBCDIC (Extended Binary-Coded-Decimal Interchange Code), på PC'ere et alfabet, hvis USA-version ses anvendt på figg. 4-9. Med de netop introducerede operativsystemer (MS-)DOS 3.30 og OS/2 erstattes dette nu af en række nye "Code Pages", af hvilke nr. 850 (Multilingual; fig 14) nok vil dække de flestes behov. Den rummer stort set de samme tegn som DS/ISO 8859-1, men også her, som i det tidligere IBM-tegnsæt, er de er placeret helt andre steder i skemaet!

Hex Digits 1st → 2nd ↓	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0		►		0	@	P	`	p	Ç	É	á	▒	└	ð	Ó	
-1	☺	◄	!	1	A	Q	a	q	ü	æ	í	▒	┴	Ð	ß	±
-2	☻	↕	"	2	B	R	b	r	é	Æ	ó	▓	┬	Ê	Ô	‗
-3	♥	‼	#	3	C	S	c	s	â	ô	ú	│	├	Ë	Ò	¾
-4	♦	¶	$	4	D	T	d	t	ä	ö	ñ	┤	─	È	õ	¶
-5	♣	§	%	5	E	U	e	u	à	ò	Ñ	Á	┼	ı	Õ	§
-6	♠	▬	&	6	F	V	f	v	å	ù	ª	Â	ã	Í	µ	÷
-7	•	↨	'	7	G	W	g	w	ç	ù	º	À	Ã	Î	Þ	¸
-8	◘	↑	(	8	H	X	h	x	ê	ÿ	¿	©	╚	Ï	Þ	°
-9	○	↓	)	9	I	Y	i	y	ë	Ö	®	╣	╔	┘	Ú	¨
-A	◙	→	*	:	J	Z	j	z	è	Ü	¬	║	╩	┌	Û	·
-B	♂	←	+	;	K	[	k	{	ï	ø	½	╗	╦	▀	Ù	¹
-C	♀	∟	,	<	L	\	l	|	î	£	¼	╝	╠	▄	ý	³
-D	♪	↔	-	=	M	]	m	}	ì	Ø	¡	¢	═	¦	Ý	²
-E	♫	▲	.	>	N	^	n	~	Ä	×	«	¥	╬	Ì	¯	■
-F	☼	▼	/	?	O	_	o	⌂	Å	ƒ	»	┐	¤	▀	´	

Fig. 14. IBM's Code Page 850 (Multilingual); den minder om det tidligere PC-alfabet, men har flere accent-bogstaver; endvidere er yen- og centtegnene flyttet og har givet plads for Ø og ø.

10. Et dansk (eller nordisk?) udvekslingsformat?

Et veldefineret udvekslingsformat for maskinlæsbare tekster ville have den fordel, at der skulle foretages færre kodeanalyser og skrives færre konverteringsprogrammer (jf. 8. Praktiske erfaringer ovenfor), end hvis alt skal kunne konverteres til alt. Formatet kunne baseres på et udvalg af ISO-tegnsættene og på SGML. Jeg er i gang med, til intern brug på forlaget, at fastlægge et sådant format og hører derfor gerne, hvis nogen har kendskab til noget allerede eksisterende. Desuden indbydes andre interesserede til at deltage i fastlæggelsen, hvorved en bredere anvendelighed kunne sikres.

Tak for hjælp!

Geir Andresen, INGRAF, har venligst stillet upubliceret materiale til rådighed. Anna Braasch, EUROTRA-DK, samt Jane Rosenkilde Jacobsen og Hanne Ruus, begge Københavns Universitet, har gennemlæst manuskriptet og bidraget med værdifulde kommentarer.

Referencer

Amsler, Robert A. (1986), Deriving Lexical Knowledge Base Entries from Existing Machine-Readable Information Sources. Manuskript, dateret 16 May 1986, til The Workshop on Automating the Lexicon, Marina di Grosseto 19.-23. maj 1986.

Andresen, Geir (1987), Institutt for Grafisk Forskning, Oslo, Trinn ved Tekstformidling. Foredragsmanuskript til EPMARKUP, konference for The European Publishers Markup User Group, Amsterdam, 19.-20. februar 1987.

Benbow, Timothy (1986), The New Oxford English Dictionary Project: An Introduction, SGML Users' Group Bulletin vol.1, nr.2, 1986:65-74.

Carroll, Lewis (1946), Alice i Æventyrland og Bag Spejlet. Oversat af Kjeld Elfelt og Mogens Jermiin Nissen, København, Thorkild Becks Forlag, 1946.

Cave, Francis (1986), Typographic Markup Techniques. Fourth Working Draft. Udkast til British Standard. PIRA, Leatherhead, Surrey, England, 15.8.1986.

Cowlishaw, M.F. (1987), LEXX - A programmable structured editor, IBM Journal of Research and Development vol.31 nr.1, januar 1987:73-80.

Descriptive Tools (1987): The DANLEX Group (Ebba Hjorth, Jane Rosenkilde Jacobsen, Bodil Nistrup Madsen, Ole Norling-Christensen, Hanne Ruus), Studies in Computational Lexicography: Descriptive Tools for Electronic Processing of Dictionary Data., Lexicographica Series Maior 20. Tübingen, Max Niemeyer Verlag, 1987.

DS 2089 (1974): Dansk Standard. 7-bit kodet tegnsæt for databehandling. København, Dansk Standardiseringsråd 1974.

DS/ISO 646 (1974): Dansk Standard. 7-bit kodet tegnsæt for databehandling. København, Dansk Standardiseringsråd 1974.

DS/ISO 8859-1 (1974): Dansk Standard. Elektronisk informationsbehandling. 8-bit kodede grafiske tegnsæt. Del 1: Latinsk alfabet nr. 1 København, Dansk Standardiseringsråd 1.9.1987.

EDO (1988): Jens Axelsen, Engelsk-dansk Ordbog, 11. udgave, København, Gyldendal 1988.

Erlandsen, Jens, og Ole Norling-Christensen (1988), A SGML-Based Lexicographical Workstation. Foredrag til COLING 88, Budapest; endnu upubliceret.

FDO (1980): N.Chr.Sørensen, Fransk-dansk Ordbog, 8. udgave ved I.-L. Dalager, København, Gyldendal 1980.

FORMEX (1985): Formalized Exchange of Electronic Publications. Standard generalized mark-up language (SGML) as described in Appendix B of the FORMEX manual, Luxembourg, Office for Official Publications of the European Communities, 1985.

INGRAF (1985): INGRAFs anbefalte markeringssystem, Oslo, Institutt for Grafisk Forskning, september 1985.

ISO 2022 (1986): International Standard ISO 2022. Information processing - ISO 7-bit and 8-bit coded character sets - Code extension Techniques. 3. udgave, Schweiz, International Standard Organization maj 1986.

ISO 8879 (1986): International Standard ISO 8879. Information processing - Text and office systems - Standard Generalized Markup Language (SGML). Schweiz, International Standard Organization 15.10.1986.

ISO/DIS 8613 (1987): Forslag til International og Dansk Standard. ISO/DIS 8613 Elektronisk informationsbehandling. Teksthåndtering. Arkitektur og udvekslingsformat for kontordokumenter (ODA). København, Dansk Standardiseringsråd 1987.

Lenders, W. (1986), A Computer Aided Study in the Semantic Function of Verbs in Philosophical Texts, foredrag ved 13. International ALLC Conference, 1.-4. april 1986, Norwich, UK.

Møller, Gregers (1987), Kun et ud af tre sætterier, De Grafiske Fag 1/87 vol. 83:8-9.

ODSS (1987): Supplement til Ordbog over det danske Sprog. Prøvehæfte. Udgivet af Det danske Sprog- og Litteraturselskab, København, Gyldendal 1987.

Petersen, Pia Riber (1984), Nye Ord i Dansk 1955-75, København, Gyldendal 1984.

RDO (1985): Jørgen og Valentina Harrit: Russisk-dansk Ordbog, 2. udgave, København, Gyldendal 1985.

RO (1986): Retskrivningsordbogen. Udgivet af Dansk Sprognævn, København, Gyldendal 1986.

Smith, Joan M. (1986), Generic Markup of Documents the Standard Way, SGML Users' Group Bulletin vol.1 nr.1 1986:9-11. Heri også bibliografi (s. 8).

Smith, Joan M. (1987), The Standard Generalized Markup Language (SGML) for Humanities, Litterary and Linguistic Computing vol.2 nr.3, 1987:171-175.

Vail, Simon (1987), The Road to Conversion, Graphic Repro vol. 7, nr. 4, april 1987:30-35.

Vejl. (1987): Vejledning om valg af tekstbehandlingssystem. København, Administrationsdepartementet 1987.

Ordbøger i Danmark: datamatstøttet leksikografi i praksis

Hanne Ruus & Dorthe Duncker
Københavns Universitet
Institut for Nordisk Filologi
Njalsgade 80
DK-2300 København S

I et lille sprogområde som det danske bruges der forholdsvis mange kræfter på leksikografisk arbejde. Den danske sprogbruger har behov for og krav på en rimeligt dækkende leksikografisk beskrivelse af sit modersmål, både af den nugældende rigssprogsnorm med dens regionale og sociale varianter og af tidligere sprogtilstande. Endvidere har den danske sprogbruger berettiget forventning om tosprogsordbøger både mellem dansk og nordiske og europæiske nabosprog, mellem dansk og de store internationalt anvendte sprog og mellem dansk og de sprog, som tales af de forskellige indvandrergrupper, der er kommet til landet i de senere år.

Man kan øge effekten af den leksikografiske arbejdsindsats ved koordinering og samarbejde. En forudsætning for samarbejde er lettilgængelige og ajourførte oplysninger i det leksikografiske miljø om, hvilke forskningsprojekter der er i gang, og om, hvem der arbejder med hvilke ordbogsprojekter.

Det leksikografiske miljø i Danmark blev kortlagt ret grundigt som led i det arbejde, der udførtes af et udvalg nedsat af Ministeriet for kulturelle anliggender i 1977. Dette arbejdes resultater, der bygger på data indsamlet i 1978 og 1979, suppleret i 1981, er fremlagt i betænkningen Vilkår for ordbogsarbejde i Danmark, 1982 - i det følgende kaldet Ordbogsbetænkningen.

I Ordbogsbetænkningen sammenfattes oplysninger om mål, tidsplaner og økonomi for større ordbogsprojekter, der var i gang i 1981. Virkningerne af Ordbogsbetænkningens anbefalinger blev i høj grad præget af, at udvalgsmedlemmerne havde meget forskellige meninger om et eventuelt koordine-

rende organ, et ordbogsråd, som da heller ikke er blevet oprettet.

De koordinerende bestræbelser blev fulgt op af Statens Humanistiske Forskningsråd med særligt henblik på anvendelse af edb-teknologi i et initiativområde: Edb for Tekst, Tale og Ordbøger (ETTO) 1982-85. Med støtte herfra arbejdede DANLEX-gruppen med beskrivelse af danske ordbogsdata til lagring i elektroniske systemer. Resultaterne fra dette arbejde er publiceret i bogen Descriptive Tools for Electronic Processing of Dictionary Data 1987. DANLEXgruppen består af: Gert Engel, Handelshøjskole Syd; Ebba Hjorth, Gammeldansk Ordbog; Jane Rosenkilde Jacobsen, Københavns Universitet; Bodil Nistrup Madsen, Handelshøjskolen i København; Ole Norling-Christensen, Gyldendal og Hanne Ruus, Københavns Universitet. I 1986 tog Statens Humanistiske Forskningsråd initiativ til et udredningsarbejde over området "Almindelig leksikografi" både vedrørende de teoretiske forudsætninger og de konkrete ordbogsprojekter. Målet for udredningen var en koordinering af aktiviteterne, både inden for forskningen og, hvor det er muligt og ønskeligt, mellem de igangværende og planlagte ordbogsprojekter. Udredningsarbejdet blev udført af DANLEXgruppen i tidsrummet januar til september 1987. En oversigt over arbejdet findes i Ruus 1987.

Siden slutningen af 1970erne er den væsentligste ændring i vilkårene for leksikografisk arbejde fremkomsten og udbredelsen af informationsteknologiske hjælpemidler til ordbogsarbejde. De elektroniske hjælpemidler letter og smidiggør ordbogsarbejdet og minimerer det trivielle kontrolleringsarbejde, men de kræver også ofte ret omfattende og detaljeret teknisk viden hos brugeren, hvis denne skal opnå de nævnte fordele.

En af konsekvenserne af en udbredt anvendelse af informationsteknologi inden for et forsknings- og praksisområde som Almindelig leksikografi er, at en hensigtsmæssig udnyttelse af de tekniske faciliteter kræver en præcis og struktureret beskrivelse af de data, de behandler. Med sådanne præcise databeskrivelser bliver det betydelig enklere for to eller flere projekter at finde berøringsflader og samar-

bejdsmuligheder. Den elektroniske lagring gør det endvidere overkommeligt at udtage veldefinerede delmængder fra store datasamlinger. Der er således teknisk mulighed for at udveksle og dele data som aldrig før. En anden basal forudsætning for samarbejde og koordination er en udbredt viden om igangværende og netop afsluttede projekter inden for området.

På denne baggrund valgte DANLEXgruppen at koncentrere udredningsarbejdet om

* indsamling af oplysninger om netop afsluttede og igangværende projekter inden for området Almindelig leksikografi, oplysninger om nuværende og ønsket teknologianvendelse, oplysninger om brug af andre datasamlinger og allerede etableret samarbejde,
* bearbejdning af de indsamlede oplysninger dels i seminarform dels i rapportform,
* formidling af oplysninger om undersøgelsens resultater.

Resultaterne af det udførte indsamlings- og bearbejdningsarbejde er fremlagt i rapporten Ordbøger i Danmark, 1987. Her skal vi sammenfatte og kommentere de resultater fra udredningsarbejdet, der giver et billede af datamatstøttet leksikografi i praksis i Danmark anno 1987.

Spørgeskemaundersøgelsen

Med skyldigt hensyn til de i Ordbogsbetænkningen anvendte spørgeskemaer blev der udarbejdet to spørgeskemaer, et for forskningsprojekter inden for området Almindelig leksikografi og et for ordbogsprojekter. Ordbogsskemaerne indeholder spørgsmål om det enkelte ordbogsprojekts leksikografiske type. De øvrige spørgsmål er koncentreret om ordbøgernes kilder - herunder om disse er maskinlæsbare, om anvendelsen af apparatur, om ønsket adgang til apparatur, om eksisterende og ønsket samarbejde med andre projekter.

Spørgeskemaerne blev sendt til alle højere læreanstalters sproginstitutter og til forlag, institutioner og personer, som arbejder med leksikografi og ordbøger. Med henblik på indsamling af viden om og brug af tekniske hjælpe-

midler blev ordbogsspørgeskemaet sendt til alle gruppen bekendte ordbogsprojekter, også fagsprogsordbøger og specialordbøger. Kun meget elementære ordbøger til skolebrug er ikke taget i betragtning.

Bortset fra enkelte kommercielle ordbogsproducenter har så godt som alle skemamodtagere, der arbejder med praktisk ordbogsarbejde, udfyldt disse.

Sammenfatningerne af oplysningerne fra ordbogsskemaerne (s. 54-82 i Ordbøger i Danmark) må derfor siges at give et ret dækkende billede af igangværende ordbogsprojekter, for så vidt angår deres art, deres anvendelse af apparatur og deres berøringsflader med andre ordbogsprojekter og datasamlinger.

Ordbogsprofiler

For at skabe overblik over, hvilke typer ordbøger der er i arbejde, blev der i spørgeskemaet stillet spørgsmål om en række leksikografiske karakteristika. Disse karakteristika er opstillet ud fra den taksonomi for leksikografiske oplysninger, der beskrives i Descriptive Tools 1987 s. 29-51. Fra de værkspecifikke oplysningstyper er specificeret følgende:

* om antallet af sprog: etsprogs-, tosprogs- eller flersprogsordbog
* om arten af sprog: nationalsprog, sociolekt, dialekt
* om synsvinkel: almensproglig eller fagsproglig og inden for det fagsproglige: enkelt- eller flerfaglig

Fra de generelle oplysningstyper er specificeret:

* etymologi (dvs. historiske oplysninger)
* fonetik (dvs. udtaleoplysninger)
* grammatik
* betydningsbeskrivelse
* citater (dvs. citater eller teksteksempler)

I skemaerne kan man således kende modersmålsordbøgerne på krydset i etsprogs-kolonnen og oversættelsesordbøgerne på krydset i tosprogs-kolonnen. I markeringen af de generelle

oplysningstyper kan man finde mulige samarbejdspartnere om de enkelte oplysningstyper: her kan man se, hvor man kan henvende sig om f.eks. udtaleoplysninger eller eksempler på ords anvendelse.

	Sprog										Oplysn. typer				
	Etsprogs	Tosprogs	Flersprogs	Nationalsprog	Sociolekt	Dialekt	Almensproglig	Fagsproglig	Enkeltfaglig	Flerfaglig	Etymologi	Fonetik	Grammatik	Betydningsbeskrivelse	Citater/tekster
Gyldendals Røde Ordbøger eng. - da., da. - fr., da. - ty., da. - sp.		x		x			x					x		x	
Dansk Etymologisk Ordbog, Gyldendal	x			x			x				x				
Gyldendals Fremmedordbog	x			x			x				x	x	x	x	
Dansk-Engelsk 10. udg., Gyldendal		x		x			x						x	x	
Engelsk-Dansk, B. Kjærulff Nielsen		x		x			x			x		x	x	x	
Italiensk-Dansk, Gyldendal		x					x						x		x
Dansk-Persisk Ordbog		x					x						x		x
Dansk-Kurdisk/Kurdisk-Dansk Miniordbog		x			x		x						x		x
Dansk-Tamilsk/Tamilsk-Dansk Ordbog		x		x			x						x		x
Ordbog over den danske Dialekt i Angel	x					x	x				x	x	x	x	x
Dictionnaire touareg-franÇais		x		x		x	x				x	x	x	x	x
DEMEP Dict. of Early Mod. Engl. Pron.	x			x								x			
Udtaleordbog over dansk rigsmål	x			x			x					x	x		
Nylatinsk Ordliste		x												x	
Persisk-Dansk Ordbog		x											x	x	x
Dansk Radiærordbog	x			x			x								
Dansk-Finsk Ordbog		x										x	x		
Dansk-Tysk Teknisk Ordbog		x		x				x		x			x	x	
Dansk-Engelsk Teknisk Ordbog		x		x				x		x			x	x	
Semantisk Rigssprogsordbog	x			x			x					x	x	x	x
Fra A til Z, Da. vendinger med eng., ty. og fr. ækvivalenter			x											x	x
Tjekkisk-Dansk Grundordbog		x		x			x						x	x	

	Sprog										Oplysn. typer				
	Etsprogs	Tosprogs	Flersprogs	Nationalsprog	Sociolekt	Dialekt	Almensproglig	Fagsproglig	Enkeltfaglig	Flerfaglig	Etymologi	Fonetik	Grammatik	Betydningsbeskrivelse	Citater/tekster
Dansk-Engelsk Ordbog, Vinterberg og Bodelsen		x		x			x						x	x	x
Da.-Fr. & Fr.-Da., Blinkenberg-Høybye		x		x			x	x					x	x	
Dansk-Hebræisk & Hebræisk-Dansk Ordbog		x											x	x	
Dansk-Russisk Ordbog		x					x					x	x	x	x
Dansk-Tysk Ordbog		x		x		x	x					x	x	x	x
Dansk-Tysk Handelsordbog		x		x			x	x	x	x			x	x	
DANTERM, terminologisk database			x	x				x		x		x	x	x	x
DANwORD, ordhyppigheder i moderne dansk	x			x			x						x		
Dansk Udtaleordbog	x											x	x		
Gammeldansk Ordbog	x	x									x		x	x	x
Supplement til Ordbog over det Danske Sprog	x			x			x				x		x	x	x
Holbergordbogen	x			x			x						x	x	x
Eftermid. personnavneordbog	x	x									x		x	x	
Eurotra-DK			x	x			x	x	x				x	x	
Fr.-Da. Ordbog o. samf.faglige termer		x		x				x		x				x	
JURTERM, Juridisk Ordbog: civilproces			x	x				x	x				x	x	x
Jysk Ordbog						x					x	x	x	x	x
Nye Ord i Dansk 1976-86	x			x			x				x		x	x	x
Ordbog over det norrøne prosasprog			x				x							x	x
Serbokroatisk-Dansk Ordbog		x					x						x		x
Polsk-Dansk Ordbog		x					x						x		x
Dansk-Fransk Teknisk Ordbog		x		x				x		x			x	x	
Ømålsordbogen						x	x				x	x	x	x	x
Gads Stribede Ordbøger da. - eng., da. - fr., da. - ty.		x					x					x	x	x	x

Teknologianvendelse

Viden om de forskellige projekters anvendelse af edb-teknologi er en vigtig forudsætning for vurdering af samarbejdsmulighederne. I spørgeskemaet blev der stillet spørgsmål om ordbøgernes anvendelse af apparatur, redigeringssystemer og om kildemateriale i maskinlæsbar form. Det fremgik af undersøgelsens resultater, at næsten tre fjerdedele af de projekter, som deltog, er indstillet på at inddrage teknologiske hjælpemidler i det leksikografiske arbejde. 70% af deltagerne anvender allerede edb, mens 22% har et ønske om at gøre det, men har endnu ikke fået stillet apparatur til rådighed eller har anskaffet det.

Den konfiguration, de fleste projekter arbejder med, består af en PC'er og en matrix-printer samt forskelligt redigeringsprogrammel. Enkelte har større maskiner (større kapacitet), kommunikationsudstyr (modem) og laserprinter.

Af den gruppe, som allerede anvender edb, arbejder næsten alle (87%) med redigeringsprogrammel. De anvendte systemer fordeler sig på fire typer: dedikerede systemer, databasesystemer, andre systemer og tekstbehandlingssystemer.

Godt en fjerdedel (28%) af ordbøgerne inden for gruppen anvender det dedikerede system COMPULEXIS. COMPULEXIS er et færdigt redigeringssystem, som kan anvendes af brugere uden indgående edb-kendskab (kap. 6 i Descriptive Tools og Svensén 1987). Systemet er under stadig udvikling i takt med kundernes ønsker og konfigureres til det enkelte projekt, så behovet for oplysningstyper (felter) og alfabeter dækkes. I øvrigt kan alle specialtegn anvendes. De vises korrekt på både skærm og print - uden koder. I praksis viser det sig dog, at der ikke altid er fuld overensstemmelse mellem tegnene på skærm og/eller print og/eller sats.

Ud fra erfaringerne med COMPULEXIS i undersøgelsen fremgår det, at systemets styrke - ud over dets avancerede tekstbehandlingsfaciliteter - blandt andet er, at det giver mulighed for struktureret redigering/indtastning, automatisk typografering, at både struktureret og typografisk version kan vises på såvel skærm som på print, at man kan søge på felttyper og -indhold, at der automatisk holdes kontrol med felters rækkefølge og indhold, at hele redaktionsprocessen

frem til færdig fotosats understøttes, og at der er vidtgående sikkerhedsforanstaltninger mod ødelæggelse/tab af data (f.eks. ved strøm- eller maskinsvigt).

Brugerne peger dog også på en række mangler. Særlig søgemulighederne virker utilfredsstillende. Hurtig søgning kan kun foretages på opslagsord. Alle andre søgninger kan kun udføres som tidskrævende batch-kørsler, og desuden kan resultatet af søgning kun udskrives på printede lister - ikke på skærm, og resultaterne kan heller ikke gemmes til senere brug i systemet. Systemet opleves som langsomt blandt andet af denne grund, men også på grund af den høje datasikkerhed (i hvert fald hvis man bruger disketter). En anden svaghed ved systemet er, at strukturering i flere hierarkiske niveauer kun i ringe grad understøttes, hvorfor reglerne for automatisk typografering kan blive så komplicerede, at de vanskeligt kan formuleres af brugerne. Desuden er skift mellem struktureret og typografisk version langsom (Descriptive Tools s. 176ff).

For brugerne af COMPULEXIS gælder det, at de gerne vil beholde systemet, men ønsker udvidelser og mulighed for databaselagring. En fjerdedel af projekterne i undersøgelsen har da netop også valgt et databasesystem som redigeringssystem. Der er tale om fire forskellige systemer: DANSTATUS, KnowledgeMan, Informix og MASTERFILE.

DANSTATUS er den danske version af det engelske informationssøgningssystem STATUS II. DANSTATUS bruges til tekstkorpus og terminologi- og ordbogsdatabase på Handelshøjskolen i København (Descriptive Tools s. 146ff, Nistrup Madsen 1987). DANSTATUS udmærker sig ved at være brugervenlig, give mulighed for dynamisk udvidelse af tegnsæt, ved ikke at have væsentlige datamængde- og datalængdebegrænsninger, ved ikke at have tomme felter og nogen fast feltlængdedefinition og ved at give mulighed for at ændre feltrækkefølge i de enkelte poster. Med hensyn til søgemulighederne kan man i DANSTATUS foretage fritekstsøgning, samtidig søgning i hele ordbasen, selektive søgninger og udskrifter (ordbogsartikler og dele af ordbogsartikler, nøgleordslister, frekvenslister og konkordans), og man kan bruge direkte kommandosprog ved søgning. Desuden er der mulighed for at danne menusystem og

særlige søgeprofiler, for at søge videre i en svarliste med poster fundet ved søgning og for at udskrive svarlister med nyt format uden at foretage ny søgning. Der er endvidere mulighed for at læse- og skrivebeskytte på alle niveauer og for at få oplysning om antallet af fundne svar, før man får svarene præsenteret.

Svaghederne ved DANSTATUS er, at der mangler validitets- og sekvenskontrol, at der ingen mulighed er for automatisk betydningsnummerering, og at der ingen mulighed er for samtidig søgning og repræsentation af data fra tekstkorpus og ordbase. På grund af sine begrænsede niveauopdelingsmuligheder og dermed manglende mulighed for afspejling af relationer mellem data kan DANSTATUS ikke betegnes som det optimale system. Det er endvidere ikke muligt at søge på feltnavne, og sorteringsfaciliteterne er meget begrænsede. Det er heller ikke muligt at tilføje nye feltnavne, uden at hele databasen må unloades og oprettes igen med ny databasedefinition. En ulempe ved arbejdet med DANSTATUS er desuden de begrænsede tekstbehandlingsfaciliteter.

KnowledgeMan er valgt efter dårlige erfaringer med DBASE-III (Jacobsen 1987). Brugen af DBASE-III viste, at systemet ikke kunne opfylde brugernes behov. Det var ikke stort nok, det havde ingen mulighed for alternativ alfabetisering, og det gav besvær med at arbejde med flere filer samtidig. Med KnowledgeMan er det derimod muligt at arbejde interaktivt med flere filer på en gang, det kan lade sig gøre at udføre fri alfabetisering, og systemet ledsages af et rimeligt stærkt og forståeligt programmeringssprog (a la COMAL). Den nyeste version har indbyggede menuer, der gør systemet let at gå til. Der er dog også ulemper forbundet med KnowledgeMan. Systemet er meget langsomt, fordi det er fortolkende, og det kræver indgående edb-kundskab af sine brugere, hvis man vil udnytte de faciliteter, der ligger i programmeringssproget - samtidig med at dette ikke er stærkt nok for den, som kan programmere.

Det tredje af de anvendte databasesystemer, Informix, er et relationelt databasesystem. Omfanget af en database i Informix begrænses principielt kun af det til rådighed stående hardware. Informix er et fleksibelt applikationssystem, som er forholdsvis nemt at anvende. Det indeholder en version af

det standardiserede søgesprog SQL (Structured Query Language). Desuden kan systemet udbygges med de uafhængige programmeringssprog C og Cobol.

Informix kan håndtere en del af de ønsker og behov, som fremkom under undersøgelsen (Widell 1987). Der kan dog ikke repræsenteres og arbejdes på flere poster ad gangen i editeringsfasen. Det kan kun ske via rapportgenerering.

Det fjerde databasesystem, MASTERFILE, giver gode søgemuligheder og kan arbejde sammen med et tekstbehandlingssystem (TASWORD). Til gengæld er begrænsningerne med hensyn til omfanget af databasen meget generende (kun 64Kb). Systemet er heller ikke kompatibelt med meget andet.

En fjerdedel af projekterne som bruger redigeringssystemer, bruger "andre systemer". Det drejer sig dels om typografiske systemer, dels om særlige løsninger udarbejdet direkte med henblik på det enkelte projekt. Den sidste løsning lader til at være den mest hensigtsmæssige set ud fra et brugersynspunkt. Brugerne af disse systemer har med en enkelt undtagelse ingen dårlige erfaringer og ingen ønsker om forbedringer. I det eneste tilfælde, hvor løsningen ikke fungerer tilfredsstillende, er programmellet fremstillet i samarbejde med et edb-firma og ikke af projektdeltagerne selv.

At denne type løsning overhovedet forekommer, må nok ses som et resultat af de gældende markedsforhold med hensyn til (manglende) udbud af leksikografisk anvendeligt software. Løsningen giver friere hænder til det enkelte projekt under arbejdet end f.eks. et system som COMPULEXIS gør. I forhold til COMPULEXIS vurderes et system som Informix som et mere forskningsfleksibelt edb-redskab med en relativ høj fortrydelsesret indbygget.

Den totale frihed får man dog først med egen programmering. Dette realiseres af i det mindste ét projekt med direkte specifik programmering til løsning af eventuelle problemer, så man ikke låses fast i arbejdet ved på forhånd at have defineret alle mulige opstående behov - hvorefter der opstår et man ikke havde forudset.

Den sidste fjerdedel af projekterne har valgt et tekstbehandlingssystem som redigeringssystem. Et enkelt projekt

bruger ikke engang tekstbehandling, men blot editoren til et programmeringssprog (PolyPascal). Denne editor har dog tekstbehandlingslignende faciliteter. Erfaringerne med den er imidlertid vanskeligt sammenlignelige med de anvendte tekstbehandlingssystemer, da ingen af disse er nævnt ved navn i undersøgelsen. Det brugerne af tekstbehandling primært savner, er repræsentation af specialtegn på såvel skærm som printer.

To af projekterne anvender editeringsprogrammet SCRIPTOR udviklet på Humanistisk edb-center, Københavns Universitet. Programmet kan håndtere en række fremmede alfabeter og disse kan repræsenteres både på skærm og printer. Brugerne udtrykker dog ønske om mulighed for struktureret inddatering a la COMPULEXIS.

Samarbejdsmulighederne

Generelt er der stor spredning inden for de anvendte systemer. Dette rejser en meget central problemstilling: kompatibiliteten. Hvis der for alvor skal være basis i fremtiden for et udstrakt samarbejde, må der ske en standardisering på både hardware- og softwarefronten. Denne problematik vedrører også udveksling af maskinlæsbart kildemateriale, som i nogen grad foreligger (Hjorth 1987). Materialet skulle også gerne være maskinlæsbart for dem man udveksler med (Norling-Christensen 1988).

Kompatibilitetsproblemet vedrører både programmel og apparatur og forekommer både i forbindelse med forskellige fabrikater og fabrikater af samme type. Man kan komme ud for, at et system kun fungerer på sin "egen" maskine, da ikke alle fabrikater er 100% kompatible endog med sig selv! Desuden kører f.eks. COMPULEXIS på enkeltstående maskiner uden fælles adgang til hverken data eller printer. Systemet kan endvidere kun (for tiden) køre på Apricot og Sirius/Victor maskiner som ikke er særligt udbredte. Det giver problemer med hensyn til service og med at bruge maskinen til andet end lige netop COMPULEXIS. Det er imidlertid planlagt, at COMPULEXIS skal blive tilgængeligt på IBM PC og kompatible maskiner.

Det meget varierede udsnit af systemer i brug er problema-

tisk i forbindelse med udveksling af data mellem projekter med forskellige systemer.

Brugere af DANSTATUS kan uden videre udveksle data indbyrdes. Sådan burde det i princippet også være for det store antal brugere af COMPULEXIS, men på grund af individuelt udarbejdede formater repræsenteres data forskelligt i systemer med forskellige formater. Hvis to eller flere projekter derimod arbejder med det samme format, skulle det være muligt for dem at udveksle data med hinanden. Udveksling af data mellem COMPULEXIS og DANSTATUS har været genstand for en undersøgelse (Vestergaard 1987) om overførsel af data fra COMPULEXIS til DANSTATUS. Det lod sig kun gøre, fordi COMPULEXIS indvilligede i at lade disketterne konvertere til magnetbånd (7-bit ASCII). Dette magnetbånd dannede i undersøgelsen udgangspunkt for forsøg med to databasesystemer: DANSTATUS og DataFlex. Det lykkedes kun at nå gennem hele proceduren for DANSTATUS' vedkommende. Med hensyn til mulighederne for udveksling af data mellem de øvrige anvendte systemer er situationen bedst for tekstbehandlingssystemerne. De fleste tekstbehandlingssystemer har en facilitet til lagring af dokumenter i ASCII-filer. En sådan fil kan siden anvendes i et andet tekstbehandlingssystem. Ulempen ved denne type overførsel er, at alle specialkoder går tabt (understregning, fed skrift osv.). Der findes hjælpeprogrammer som kan oversætte direkte mellem forskellige tekstbehandlingssystemer så alle oplysninger bevares, men da de anvendte systemers identitet ikke fremgår af undersøgelsen er det vanskeligt at anbefale et bestemt oversættelsesprogram.

Undersøgelsen giver ikke noget entydigt billede af det optimale system. Brugerne af henholdsvis COMPULEXIS og DANSTATUS ønsker udvidelser med det andet systems styrkesider. Således ønsker COMPULEXIS-brugerne mulighed for databaselagring, mens DANSTATUS-brugerne ønsker øgede tekstbehandlingsfaciliteter. Denne tendens afspejles ikke blandt ønskerne fra brugerne af egentlige tekstbehandlingssystemer og de øvrige databasesystemer. Således giver kun enkelte tekstbehandlingsbrugere udtryk for et ønske om at kunne lagre i database, ligesom databasebrugerne ikke formulerer ønsker om adgang til tekstbehandling. Udveksling af data mellem forskellige

databasesystemer kan vise sig at være mere eller mindre problematisk alt efter, hvilket format systemernes datastruktur kan normaliseres til.

Generelt for alle de anvendte systemer gælder det, at det ville hjælpe, hvis man får udviklet noget programmel, som kan oversætte alle typer data til ét fælles format. Her vil et SGML-format (Standard Generalized Markup Language) nok være det bedste at udveksle data i (Descriptive Tools s. 139-144, Smith 1987, Norling-Christensen 1988).

Referencer

Descriptive Tools for Electronic Processing of Dictionary Data, Studies in Computational Lexicography; The DANLEX-group: Ebba Hjorth, Bodil Nistrup Madsen, Ole Norling-Christensen, Jane Rosenkilde Jacobsen, Hanne Ruus; Lexicographica, Series Maior, Band 20, Max Niemeyer Verlag, Tübingen 1987.

Hjorth, Ebba: Materialesamlinger, i Ordbøger i Danmark; s. 98-102.

Jacobsen, Bent Chr.: Den Arnamagnæanske Kommissions Ordbog, i Ordbøger i Danmark; s. 150-152.

Nistrup Madsen, Bodil: Dansk-fransk Ordbogsbase, i Ordbøger i Danmark; s. 124-131.

Norling-Christensen, Ole: Læsning af maskinlæsbare tekster, i LAMBDA No. 7. København 1988.

Ordbøger i Danmark, En oversigt, DANLEXgruppen, København 1987.

Ruus, Hanne: Det udførte arbejde, i Ordbøger i Danmark; s. 2-6.

Smith, Joan M.: The Standard Generalized Markup Language (SGML) for Humanities Publishing, i Literary and Linguistic Computing, Vol. 2, No. 3, Oxford 1987; s. 171-175.

Svensén, Bo: Handbok i lexikografi, Esselte Studium TNC, 1987; s. 260-64.

Vestergaard, Bodil: Undersøgelse af databasesystemer til ordbøger, LAMBDA No. 2, Institut for Datalingvistik, Handelshøjskolen i København, København 1987.

Vilkår for ordbogsarbejde i Danmark, Betænkning afgivet af det af Ministeriet for kulturelle anliggender nedsatte ordbogsudvalg, Betænkning nr. 967, København 1982.

Widell, Peter: Databasesystemer i ordbogsarbejde - eksemplificeret ved det relationelle databasesystem Informix, i Ordbøger i Danmark; s. 132-149.

Nordisk seminar om datamatstøttet leksikografi og terminologi.
Handelshøjskolen i København. 5. og 6. november 1987.

TERMINOLOGI, ARBEIDSINSTRUKSER OG LAGERSTYRING - OM KODEUTTRYKK I FAGSPRÅK

Ivar Utne,
Nordisk institutt, Avd for norsk leksikologi
og
Norsk termbank,
Universitetet i Bergen, Norge

1 Innledning

Emnet for denne artikkelen er ikke-ordinære språkuttrykk, som her vil bli kalt kodeuttrykk. I hovedsak dreier det seg om korte informasjonsmettede uttrykk. Disse uttrykkes i en språkform som gjør dem vanskelige å forstå. Uttrykkene er særlig vanskelig å forstå for dem som ikke kjenner denne delen av fagspråksbruken innenfor de emneområdene kodeuttrykkene hører til.

Formålet med denne artikkelen er å se på form, funksjon, samt terminologisk og informasjonsteknologisk status for kodeuttrykk som brukes til å beskrive og systematisere arbeidsoppgaver og gjenstander.

Eller mer spesifikt:

- Dette er en redegjøre for bruken av kodeuttrykk i fagspråk, dvs symboler, forkortelser/kortformer og systematiske uttrykk, dvs presentasjon av uttrykkenes:
  - form: siffer, bokstaver, kortformer, ikke-alfabetiske tegn, kombinasjoner av tegn og uttrykk, og systematikk
  - funksjon: effektivisering, entydighet, språkuavhengighet o a
  - bruksområder: arkiv, lagerstyring, administrasjon, instrukser o a

- Jeg skal ta opp bruken av kodeuttrykk i fagterminologisk sammenheng, spesielt i:
  - termpostformat: felttyper i ordboksformat (som term, forkortelse, definisjon o a), klassifisering av begreper
  - dokumentsystemer: informasjon i kompakt form, entydige gjenfinningskriterier, kopling mellom innholdsområder/-felt

2 Beskrivelse

I denne framstillingen skal jeg ta opp fagspråklige uttrykk som ikke kan regnes som ordinære ord, dvs kodeuttrykk.

Med ordinære ord menes her:

- bokstavsekvenser satt sammen etter naturlig språks regelmessige orddanningsprinsipper, primært det hjemlige nasjonalspråket.

Disse orddanningsprinsippene baserer seg på sammensetninger med ordinære røtter, prefikser, suffikser, sammensetningsfuger og bøyingsendelser.

Denne orddanningen skjer naturlig, dvs uten systematisk styring, eventuelt med filologisk kontroll.

Jeg vil dele kodeuttrykk, eller ikke-ordinære språklige uttrykk, inn i tre hovedtyper.

i Kodeuttrykkene kan være dannet på grunnlag av lengre språklige uttrykk (ett eller flere ord):

- forkortelser - dvs vidt aksepterte eller standardiserte
- kortformer - dvs laget ad hoc og brukt i avgrensete miljøer

Kortformer, som normalt ikke er vidt kjente, bør helst dannes etter fastlagte prinsipper, slik at en lettere kan tolke dem uten dokumentasjon. Prinsipper kan være:

- ordstart (en eller et fåtall bokstaver), ofte bare konsonanter:
  fl for flaske og p for program
- ordstart i alle ord eller ledd i det fulle uttrykket:
  skr for sentralkontrollrom
- ordstart og ordslutt, ev avgrenset til konsonanter:
  vn for vann og msk for menneske
- alle eller et utvalg konsonanter:
  mskn for maskin og prg for program
- delvis fontisk gjengivelse:
  xqt for execute og GFX for Gullfaks
- blanding av prinsipper:
  kjvsk for kjølevæske

ii Kodeuttrykkene kan være uavhengige av andre språklige uttrykk:

- bokstavkoder (som ikke er forkortelser av lengre uttrykk)
- sifferuttrykk
- ikke-alfabetiske tegn, f eks greske bokstavtegn og utbredte symboler, som &, /, %, _, $ og £.
- tegninger, som på veitrafikkskilt, skilt på flyplasser og faremerking på emballasje

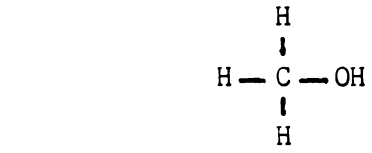
iii Kodeuttrykkene kan være systematisk oppbygd av enkeltledd som hvert har betydningsinnhold og slik at hele uttrykket er produktet av disse.

Grunnlaget for systematikken er sluttede sett og regler for sammensetning eller plassering av leddene.

a Leddene inngår i et sluttet sett, som kan bestå av følgende typer:

- forkortelser og kortformer
- språkuavhengige uttrykk
- konstruerte ordlagingselementer, f eks i kjemiske termer: para, an, ol, sid o a.

Et stoff som i norsk dagligtale kalles tresprit, har den standardiserte fagspråklige termen metanol (engelsk: methanol). Den kjemiske strukturformelen, som ser ut som en figur, men som også brukes i tekst, er:



Dette kan helt entydig omdannes til den kjemiske formelen som uttrykker at det er knyttet tre hydrogenatomer og en OH-gruppe til ett karbonatom:

CH_3OH

Ifølge regler for danning av kjemiske fagtermer utabeidd av International Union of Pure and Applied Chemistry (IUPAC) kan kjemiske forbindelser omsettes til en term, også kalt systematisk navn, ved hjelp av konstruerte ordlagingselementer og regler for hvordan de skal kombineres (se nedenfor). (En innføring i kjemisk nomenklatur eller termdanning er f eks Ringnes 1984.) I dette tilfellet er følgende tre ordlagingselementer med angitt betydning aktuelle:

met = ett C-atom
an = bare enkeltbindinger, dvs bare én strek fra C-atomet til hver av de tre H-atomene - det kunne vært flere
ol = alkoholgruppe, dvs OH-gruppe, knyttet til C-atomet

Et annet eksempel finner vi hos den internasjonale lufttransportorganisasjonens (IATA) som klassifiserer

konstruksjonen av fraktemballasje etter emballasjeform og materiale, slik at f eks emballasjeformene 1 er tønne og 4 er kasse, mens f eks materiale A er stål, C er naturlig tre, D er finér og H er plast. Med grunnlag i dette er 4C trekasse og 1A ståltønne. (IATA 1988)

b Leddenes plassering kan være bestemt av regler, som kan foreskrive:

- posisjoner
- rekkefølge
- obligatoriske og ikke-obligatoriske ledd
- avhengighet mellom ledd

Tilsammen innebærer dette strengt standardisert ordvalg og syntaks, der det ikke nødvendigvis er mellomrom mellom ordene.

Merking av tekniske gjenstander (utstyr, instrumenter og rør) gjøres ofte slik at hver av de fire hovedelementene i 21-PDSLL-025-A (eksemplet er konstruert etter forbilder fra oljeindustrien) som betegner et instrument, skal tolkes som systemnummer (innenfor et overordnet system), funksjon, sekvens (mindre del av systemet) og kode for å skille fra hverandre samme type instrumenter i samme rørkrets. Uttrykket PDSLL som står for det engelske Pressure Differential Switch Low Low kan tolkes etter en standard fra Instrument Society of America (ISA 1973) som foreskriver at en P i første posisjon står for pressure, at D står for differential og modifiserer leddet foran, at S utover første posisjon (og under visse betingelser) står for switch, og at L-ene står for low, som angir bryterstilling. Norsk term for PDS er differansetrykkbryter. Uttrykket PSV kan ifølge den samme standarden tolkes som Pressure Safety Valve, som innebærer at S som modifiserende ledd står for safety. Norsk term er trykkstyrt sikkerhetsventil.

Istedenfor systematikk brukes ofte kun sekvensiell nummerordning, der begrepet bak hvert nummer er definert. Slike nummer er altså å betrakte som uttrykk uavhengige av andre språklige uttrykk slik det ble framstilt ovenfor.

3 Bruksområder

Danning og bruk av kodeuttrykk kan være planlagt og systematisk eller impulsiv og usystematisk. Bruksområdene er utallige yrkesfaglige sammenhenger der språkbruken krever eller kan effektiviseres med bruk av koder framfor naturlig språk. Behovet styres særlig av plasshensyn, effektivisering av skriving, systematikk og automatisk databehandling. Sentrale bruksområder er:

- bibliotekssystemer:
  - Dewey og andre system med desimalkoder eller kombinasjoner av bokstaver og siffer

- arkivnøkler/-systemer:
  - f eks arkivnøkkel for offentlige kontorer

- telefontjenester:
  - bruk av siffer- og tegnkombinasjoner til å kommunisere med bl a televerket, tjenesteytende bedrifter og banker

- vare- og lagerkataloger:
  - elnummer for merking av alt elektrisk utstyr (godkjent markedsført i Norge), MESC som er en utstyrsnummerering brukt i oljebransjen (utviklet av Shell)

- varemerking:
  - strekkoder for vareidentitet; farge- og stoffnummer

- stofflister:
  - for kjemi: dels nummersystemer hovedsaklig bygd på sekvensiell nummerering, som CAS-nr (Chemical Abstracts Summary), UN-nummer (FN-nummer), EF-nummer, og systematisk oppbygde kodeuttrykk i kjemiske formler og systematisk oppbygde stoffnavn

- utstyrsmerking:
  - dimensjoner, materiale, plassering, funksjon, tilkoplingssted

- merking for tjenesteyting:
  - bestilling av middag på en kafe, f eks med koder for bordnummer, rettnummer, og med eller uten forrett, dessert og kaffe
  - merking for transport av gods eller post, f eks flyplasskode (3-bokstavskode som f eks CPH for København), flykode (selskapskode, som SK for SAS, og rutenummer), sikkerhetskode (faretype, behandlingsrestriksjoner)
  - instrukser med angivelse av sted, hjelpemidler og type aktivitet, se neste pkt

- instrukser, med betingelser eller tilstander som er forutsetninger for eller som krever visse handlinger (aksjoner) uttrykt i kodeform:
  - tilstanden kan være alarmsignal eller måldiagnose som spesifiserer sted og type mangel
  - handlingen kan foreskrives med sted og type handling

- meldinger:
  - skiltmerking på maskiner og kontrolltavler, veitrafikkskilt og informasjonsskilt for allmennheten (som røyking forbudt-skilt)

- merking for automatisk dataregistrering:
  - strekkoder og bestemte skrifttyper eller tegn for optisk lesing; tidligere hullkort
  - brukt for giroblanketter, prislapper og tippekuponger

4 Formål

Formålet med bruk av kodeuttrykk er å oppnå effektivisering av

kommunikasjon og informasjonsbehandling. Denne språkbruken har vokst opp med bruken av skjemaer og arkivsystemer, og satt ny fart gjennom kontorautomasjonen, dvs bruk av edb i kontorsektoren. De aktuelle aktivitetene krever streng språkstandardisering, som her vil si entydig definerte begreper med entydige uttrykk, som kan kalles termer, men som ofte skiller seg sterkt ut fra hva vi til vanlig forstår med termer.

Under forutsetning av at den menneskelige arbeidskraft ikke fungerer mer effektivt med naturlig språk enn hva de selv eller maskiner gjør med konstruerte koder, så oppnås effektivisering gjennom muligheter for:

- automatisering
- entydighet
- klassifikasjon av informasjon (som også er en forutsetning for automatisering)
- begrensing av dokumentmengde, informasjonsmedier og informasjonsflate gjennom plassøkonomi.

Automatisering vil si at en forenkler arbeidsprosessen ved:

- bruk av datamaskiner
- mekaniske maskinløsninger, som f eks sorterer eller klassifiserer etter fysisk form
- sterkt rutinepregede oppgaveløsningsmetoder for mennesker

Entydighet vil si at:

- hvert begrep har:
  - et klart definert betydningsinnhold
  - at hvert begrep har ett eller et klart avgrenset sett av uttrykksmuligheter.

- hvert uttrykk har:
  - en entydig og veldokumentert danningsmåte
  - mulighet for reversering fra kodeuttrykket, enten til sine enkelte meningsbærende komponenter eller til et lengre språklig uttrykk

Entydig klassifikasjon oppnår en også gjennom standardisering, som igjen fungerer best dersom den er bygd opp systematisk, med:

- ett uttrykk for samme begrep eller innhold (som ovenfor)
- ordning av begreper eller innholdskategorier i grupper og undergrupper
- kategorisering som vises gjennom formen, dvs tolkbare enkeltledd samt kjente kombinasjonsregler gjør det mulig å forstå den samlete betydingen og hvor i et større system et kategorinavn hører til

En entydig og standardisert klassifikasjon og uttrykksmåte fungerer best dersom mulighetene for uheldig sammenblanding med andre språklige uttrykk er liten både for mennesker og maskiner. Det kan bl a bety preferanse for koder som ikke er avledet av naturlig språk når begrepene ikke er synonyme med ev lignende begrep brukt i naturlig språk. Men det forhindrer ikke at forkortelser og kortformer kan brukes når det er synonymi mellom kode og naturlig språk.

Plassøkonomi vil si at kommunikasjons-, behandlings- eller lagringssituasjonen er slik at plassbegrensning enten er en gitt rammebetingelse eller vurderes som fordelaktig av aktørene. Viktige bruksområder er tekster på tegninger, merking av teknisk utstyr og kontrolltavler, hvor det i praksis er lite rom for prosa, bl a fordi en ønsker at ustyret eller dokumentene har minst mulig fysisk omfang pga vekt og oversikt. I databehandling reduserer tekstmengden både lagrings- og behandlingstid, samtidig som kortere uttrykk (muligens) lettere kan skrives entydig enn lange ytringer, og kortere uttrykksmåter gir mer informasjon på liten plass og dermed gi bedre mulighet for oversikt under innskriving og utskrift på skjerm og papir. Men dette må naturligvis veies opp mot vanskene både med å skrive og lese kodeuttrykk.

Utforming av begrepssystemer og språklige uttrykk, inkl kodeuttrykk, er først og fremst språklig arbeid, men svært ulikt hva filologer og språkkonsulenter vanligvis driver med. Dette er et stort språkbruksområde hvor utviklingen svært ofte er uten kontroll, og hvor det er stort behov for skikkelig språkplanlegging og -styring. Og dette er det er ofte tilløp til innenfor enkelte fag, som i kjemi, transport, merking av teknisk utstyr, prosjekteringsbeskrivelser, biblioteks- og arkivvesen. Men den mest utbredte praksisen er mange konkurrerende systemer innenfor samme fagområde.

5 Motforestillinger

Det fins minst tre hovedtyper ulemper med bruk av kodeuttrykk, som i noen grad kan ha relevans dersom uttrykkene brukes utover fagspråklige kontekster eller tar unødig overhånd i yrkeslivet.

1 Kodene er uforståelige og fremmede:

- Kodene er allment uforståelige, eller kryptiske, som fargenummer på varedeklarasjoner, varenummer, elnummer for klassifikasjon av elektrisk utstyr.
- Kodene er ekskluderende, dvs fungerer som gruppesjargong.
- Dokumentasjonen kan mangle eller være vanskelig å skaffe.
- Avstanden fra naturlig språk øker muligheten for skrive- og lesefeil.

2 Begrepsinndelingen samsvarer iblant ikke med den virkeligheten som skal karakteriseres:

- Systemene kan være for begrenset til å fange akutelle begreper etter hvert som det kommer til nytt materiale

som skal klassifiseres innenfor aktuelle emneområder.

3 Kodene tilfredsstiller ikke krav til naturlig og god språkbruk:

- Veksten/utviklingen i teknisk terminologi generelt og i forkortede uttrykk spesielt, er et slags ugras-syndrom (dvs uønsket), fordi den ikke følger vanlige språkregler og består ikke av ordinære språktegn, jf forkortelser, koder og siffer.
- De består ofte av internasjonalt, ikke hjemlig språkmateriale.
- De er ofte matematiske, og ikke språklige.
- De representerer en utvikling av kunstig og konstruert språk som i seg selv er fremmedgjørende.
- De verken styres eller kontrolleres av fagfolk med tradisjonell språkkompetanse.

6 Terminologisk klassifisering og format

6.1 Terminologisk status

Den terminologiske statusen for disse uttrykkene kan betraktes som forkortelse, symbol, term, frase, uttrykk som er resultat av orddannings- og syntaksregler, definisjon og figur. Hvert uttrykk eller hver kode kan være en eller flere av de nevnte typene fordi en formell og en funksjonell klassifisering fører til ulike resultater. Uttrykkene kan, dersom en tar begge klassifikasjonsmåter med, være:

- forkortelse, fordi
  - de formelt kan være dannet ved forkorting av lengre ordinære språklige uttrykksmåter

- symbol, dvs internasjonalt standardisert kode, som bokstaver, siffer, tegning eller andre tegn, fordi
  - de formelt er internasjonal standard

- term, dvs ordinært språklig uttrykk, fordi
  - de funksjonelt sett faktisk brukes i tekst
  - de funksjonelt sett kan brukes effektivt som benevnelse, dvs har slik form (lengde, mulig gjengivelse); tegninger fungerer f eks like godt som eller bedre enn ordinære språklige uttrykk på veitrafikkskilt og skilt henvendt til publikum på flyplasser o a
  - det i gitte kontekster og for visse begreper formelt sett ikke fins ordinære ord eller uttrykk som uttrykker samme meningsinnhold

- frase, dvs et en relativt fast ordrekkefølge, som vi her kan tøye til også å omfatte sammenskrevne ledd med eget avgrenset meningsinnhold, fordi

  - en del kodeuttrykkene har etablert seg som faste uttrykk uten at en lenger legger særlig vekt på leddenes innhold isolert sett

- uttrykk som er resultat av syntaks- og orddanningsregler, dvs regler for hvordan språkelementene inngår i større enheter, fordi

  - det fins kodeuttrykk som er dannet av andre kodeuttrykk, inkl forkortelser og kortformer

- definisjon, dvs innholdsbeskrivelse med grunnlag i predefinerte eller allment forståtte uttrykk, fordi

  - denne typen uttrykk ofte formelt og funksjonelt sett er entydige uttrykk som karakteriserer innhold og bygger på uttrykk med predefinert betydningsinnhold, slik det er best kjent i kjemiske formler

  - uttrykkenes form funksjonelt sett, trass i at de ikke er ordinære språklige uttrykk, ikke er til hinder for presisjon av innhold og således fungerer begrepsavklarende, som figurer (tegninger), tabeller (matriser) der visse felt i tabellen inngår i begrepet, og systematiske uttrykk eller formler

  - definisjoner funksjonelt sett kan være både snittmengder og unioner mellom mengder, og kan uttrykkes i formler og tabeller som:

    - en mengde som er summen av mengder framstilles ved oppramsing av ord, uttrykk, koder, figurer; matematisk framstilles dette som unioner mellom mengder

    - en mengde som er skjæringspunktet mellom mengder kan framstilles som tekst i formen en X som har egenskapen Y, eller som kryssningspunkter tabeller, eller matematisk som snittmengder

  I omtalen av IATAs emballasjekoder ble det vist hvordan emballasjetypen med koden 4C for trekasse kom fram med kombinasjonen av eller snittet mellom de to mengdene 4 og C, altså de gjenstander som er kasser og er av tre.

  IATA har også koder for emballasjetyper klassifisert etter hva de tåler, eller rettere etter hva de kan godkjennes for. Det vil si hvor sterke de er med hensyn til støt, lekkasje o a. Hvert eneste stoff som det er forbundet med sikkerhetsrisiko (brann, forgiftning) å sende, har fått tilordnet en bestemt emballasjeklasse. Disse emballasjeklassene, som kan betraktes som begreper,

betegnes med et navn eller en kode. Metanol skal f eks transporteres i emballasjeklasse nr 305 som ytre emballasje under visse forutsetninger. Emballasjeklasse nr 305 er definert som unionen av flere emballasjetyper, deriblant 4C (trekasse), 1A (ståltønne), og 1H (plasttønne).

- figur, dvs tegning eller annen form for visuell framstilling, f eks tabell,

  (Karakteristisk for en figur er at det er en visuell framstilling der bruken av språklige tegn, siffer eller symboler i seg selv ikke er det dominerende for formidlingsformen.)

  fordi:
  - uttrykkene har klart ikke-språklig form enten som strektegninger, som
    - symboler for div utstyr på tekniske tegninger, og som fungerer som termer
    - uttrykk for strukturer (bl a den kjemiske strukturen slik det ble vist for metanol ovenfor)
    - tabeller som klargjør den mengden som kan utledes av sett og snittmengder, og som fungerer som definisjoner (som i eksemplet nedenfor)

  En figur som klargjør begrepsinnholdet i IATAs emballasjeklasse nr 305 kan se slik ut:

		stål A	alumin. B	tre C	finer D	formtre F	fiber G	plast H	tekstil L	papir M
tønne	1	x	x		x		x	x		
(..)	2									
kanne	3	x						x		
kasse	4			x	x	x	x			
veske	5									
komb.	6									
trykkbh.	7									

Figur 1. IATAs emballasjeklasse nr 305

6.2 Synonymi, koreferanse og kompatibilitet

Da uttrykkene brukes i sammenhenger der det er behov for entydighet både systeminternt og mellom systemer, er det også behov for at ulike systemer er kompatible, dvs slik at kodeuttrykk i forskjellige systemer refererer til samme forekomster eller referenter.

Dels kan det dreie seg om fullstendig synonymi slik at innholdet, dvs både intensjon (begrepsklassifikasjonen særlig med hensyn til plass i et videre begrepssystem) og ekstensjon (de faktiske forekomstene av gjenstander, prosesser eller egenskaper det refereres til), er det samme. Og dels kan det dreie seg om

koreferanse, slik at kodene refererer til samme ekstensjon (forekomster) uten at de er underlagt samme intensjon (samme begrepsavgrensing). Koreferanse er aktuelt når ulike systemer refererer til samme forekomster, men fra forskjellige perspektiver.

Kompatibilitet eller ekvivalens mellom klassifikasjonssystemer trenger ikke innebære ekte synonymi, det kan være tilstrekkelig med koreferanse. Slik kan koreferente uttrykk være del i flere klassifikasjoner eller begrepsstrukturer. Og på den måten kan begrepsdefinisjonene i de ulike systemene er ulike trass i at referentene er de samme. Således kan samme kjemiske stoff inngå i systemer der begrepskjennetegnene, som er bestemt av formålet, primært kan være så forskjellige som: struktur, faretype (brann, gift o a), tiltakstype ved brann eller utslipp, bruksformål (-område), transportkrav, fabrikant o a.

6.3 Termpostformat

Denne informasjonen skal helst inn i en termposts truktur. En slik struktur bør ta vare på følgende hensyn:

- kunne vise relevante klassifikasjoner for brukerne, slik at mens en filolog kanskje vil legge vekt på formelle trekk som fullstendige språklige uttrykk og bruke det som hovedoppslag, så vil dokumenforfattere heller ha et hovedoppslag som er tilpasset den akutuelle språkfunksjonen, f eks skiltproduksjon eller merking av tegninger

- prioritere funksjon framfor form i tråd med brukerbehovene, dvs vise hvilket uttrykk som primært brukes i hvilken kontekst selv om det ikke har tradisjonell språkform - det står f eks ikke samme tekst eller symbol på et dørskilt som det vil stå i arkitekttegningen for samme rom

- samle alle uttrykk for samme begrep

- vise hva som er standardiserte
  - uttrykk
  - definisjoner

- vise alternative men ikke standardiserte
  - uttrykk
  - definisjoner

- dokumentere bruk av former og ev definisjonsvarianter veksler med kontekst; formatet må kunne brukes til å strukturere informasjon om ev brukssituasjon/kontekster for de aktuelle formene

- dokumentere, som videreføring av forrige punkt, koreferanse, dvs hvordan samme uttrykk som betegner samme fysiske gjenstand eller prosess inngår i ulike formålsbestemte begreps- eller klassifikasjonssystemer

- kunne gjenspeile begrepsstruktur, særlig
  - over-, under- og sideordnete begreper

og ev andre semantiske relasjoner som
  - tidsfølge, årsakskjede, del-helhet, gjenstand-til-prosess

- vise hvordan uttrykket forholder seg til fraser eller danningsregler for uttrykk, dvs hvordan
  - det er satt sammen av enkeltelementer
  - med hvilke regler det kan inngå i større strukturer eller uttrykk

- vise historikk (dvs logg for endringer) og ev lokal standardisering, dvs i tid og rom

- referanse, for ytterligere informasjon i
  - tekst og databaser
  - figurer
  - kompetansemiljøer

7 Dokumentasjonssystemer - kodeuttrykkenes og systematikkens formål og konsekvens

Den termiologiske klassifikasjonen og skissen av termpostformat er et tjenlig grunnlag i utvikling av et fullstendig dokumentasjonssystem. Jeg skal kort skissere hovedkomponenter i et dokumentasjonssystem og antyde hvordan temaer fra framstillingen så langt kan passes inn i et slikt system.

Et dokumentasjonssystem er et system for oppbevaring, gjenfinning og behandling av informasjon i form av dokumenter, grafikk og andre datatyper.

Informasjonen i et slikt system kan rapporteres, dvs skrives ut, i forskjellige formater og med forskjellige tilleggsberegninger, avhengig av formål og perspektiv.

Systemene struktureres via pekere mellom informasjon som kan eller skal knyttes sammen. Disse pekerne kan ofte knytte sammen:

- synonyme eller nærsynonyme begreper eller koreferente uttrykk, f eks
  - beskrivelse, figur, term, lagernummer, annen kode, uttrykk for kompatible eller ikke-kompatible begreper, og lengre tekster

- begreper som inngår i en organisert struktur der begrepene eller informasjonsenhetene kan ha knyttet til seg koder, tallinformasjon, figurer eller større dokumenter, f eks i klassifikasjonssystemer med over- og underordnete begreper, i tabeller med del-helhetsstruktur, eller prosessstyrings- eller administrasjonssystemer med f eks gjenstand-prosess- og prosess-til-prosess-relasjoner o a; i en utskrift eller som brukermelding kan koder blåses opp til mer eksplisitt informasjon (tekst, figurer)

- begreper hvorav det ene er en egenskap ved det andre, f eks
  - bruksmåte, farekode (gift, brannfare osv)

Utviklingen går klart i retning av dokumentasjonssystemer. Og systematikk, entydighet og korte uttrykksmåter, oftest som kodeuttrykk, kan så langt se ut til å være viktige forutsetninger, eller den prisen vi må betale for å kunne samle å finne fram i store informasjonsmengder.

Disse systemene kan også romme tekster med naturlig språk som er koplet med synonyme kodeuttrykk. Slik kan systemene når de har funnet akutell informasjon eller beregnet seg fram til et svar, presentere dette for brukerne i relativt naturlig språk. Når brukerne skal gi meldinger elle stille spørsmål til systemene kan naturligvis brukerne også få spørsmål og alternativer som naturlig språk og ikke bare som kodeuttrykk. Mye av dette ser vi allerede i informasjonsystemer med et begrenset antall dialog-alternativ.

Erstatning av kodeuttrykk med mer naturlig språk krever uktvikling av såkalt kunstig intelligens. Og en viktig forutsetning for dette er godt gjennomarbeidde begrepssystemer, oversikt over synonymer, og språkregler som beskriver hvordan ord og uttrykk sammen gir et komplekst meningsinnhold. Dette er også forutsetninger for at nåtidas systemer med mye kodeuttrykk skal kunne fungere effektivt, hvilket har vært hovedtema i denne artikkelen.

8 Litteratur

IATA 1988 = International Air Transport Association, 1987. Dangerous Goods Regulations. 29th Edition. Effective 1 January 1988. Montreal - Geneva. ISBN 92-9035-109-8

ISA 1973 = Instrument Society of America, 1973. ISA Standard S.5.1 rev. 1973

Ringnes, Vivi, 1984. Hvordan sette navn på kjemiske forbindelser. Oslo. ISBN 82-02-10003-8

SIMULERING AF RELATIONEL DATABASE

Bodil Nistrup Madsen
Institut for Datalingvistik

Indlæg på Symposium for datamatstøttet leksikografi og terminologi, 5.-6. november 1987, Handelshøjskolen i København

I det følgende rapporteres om et forsøg, som er gennemført med henblik på at afprøve, om et informationssøgningssystem med forholdsvis begrænsede datastruktureringsmuligheder kan bringes til at fungere således, at man opnår de samme fordele som i et relationelt databasesystem.

Forsøget er beskrevet i detaljer i et særskilt LAMBDA-nummer, LAMBDA Nr. 6 (Nistrup Madsen 1988), hvorfor indlæggets indhold her gengives i forkortet form uden oplysninger og eksempler af system- eller programmeringsteknisk art.

Jeg vil gerne takke mine kolleger i DANLEX-gruppen, uden hvis opmuntring og støtte forsøget ikke kunne gennemføres. En speciel tak til Hanne Ruus for gode forslag og til Ebba Hjort, som har leveret eksempelmateriale.

1. BAGGRUND

Ordbogsartikler i videnskabelige ordbøger indeholder ofte et meget stort antal informationstyper, som indgår i forskellige relationer med hinanden. Ved edb-behandling af leksikografiske data, f.eks. ved lagring i et databasesystem, skal de logiske forbindelser mellem de forskellige oplysninger afspejles, således at relationerne kan anvendes ved søgning og præsentation af data.

Ved Handelshøjskolen i København har man i en årrække arbejdet med systemet DANSTATUS til forskellige terminologi- og ordbogs-

projekter. DANSTATUS er den danske version af det engelske informationssøgningssystem STATUS II. I forbindelse med DANLEX-gruppens projekt "Lagring og behandling af maskinlæsbare leksikografiske data i databasesystemer" blev der gjort forsøg med lagring af data fra en videnskabelig ordbog, Gammeldansk Ordbog i DANSTATUS. Forsøget er beskrevet i Vestergaard (1987) og konklusionen er, at DANSTATUS ikke kan betragtes som et ideelt system, da afspejlingen af relationer mellem data ikke umiddelbart er mulig. Det konkluderes endvidere, at den systemtype, der skal bygges videre på, må være relationel.

DANSTATUS har imidlertid en række fordele, som er så vægtige, at det er interessant at undersøge, om man ved hjælp af nogle særlige programmeringsfaciliteter (macrofaciliteter) i DANSTATUS kan simulere en relationel database og derved opnå den ønskede strukturafspejling.

2. ARTIKELSTRUKTUREN I GAMMELDANSK ORDBOG (GLDO)

Som et led i projektet "Edb-behandling af videnskabelige ordbogsdata" har DANLEXgruppen udarbejdet en taksonomi til klassificering af leksikografiske data. Denne taksonomi beskrives i Descriptive Tools for Electronic Processing of Dictionary Data (1987). På basis af taksonomien er der udarbejdet et format til GLDO, som er anvendt ved indtastning af en række artikler ved hjælp af ordbogsredigeringssystemet Compulexis.

GLDO-formatet er inddelt i 4 afsnit:

I: identifikationsafsnit
B: bøjningsafsnit
S: semantisk afsnit
E: etymologisk afsnit

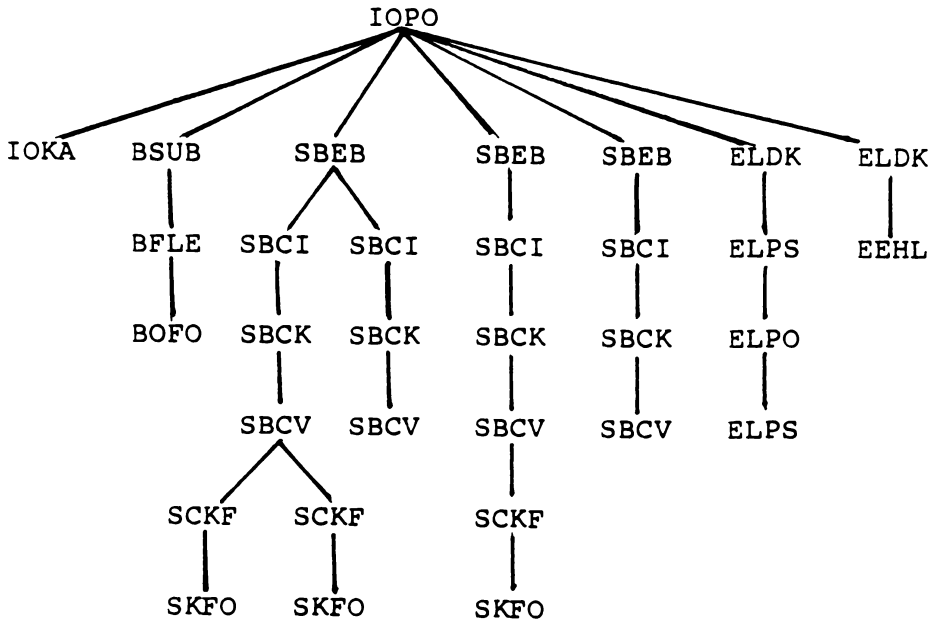
Formatet er beskrevet i Descriptive Tools (1987) og Ruus (1988). Til det her beskrevne forsøg anvendtes kun S-afsnittet, jfr. LAMBDA Nr. 6.

Nedenfor vises til eksemplificering en ordbogsartikel fra GLDO for opslagsordet "dræk", figur 1.

IHOM	IOPO dræk
IOKA	sb.
SEC	§
BSUB	n.
BØKK	Kv Rosg.
BØKV	76,15
SEC	§
BFLE	sg. bek.
BØFO	-ket
SEC	§
SBEB	snavs, smuds, skarn, spec. om ekskrementer
SBCI	ther efter skulle han al thenne verldens lyst vyrde sosom drek
SBCK	Suso.
SBCV	50,8
SCKF	træk
SKFO	Sv.
SCKF	stercus
SKFO	Lat. (jf. stercora.Filip.3,8(Vulg.))
SBCI	then indwolff, som vdsender eller vd skyuder drecket ok skarnet aff mennisken
SBCK	Kv Rosg.
SBCV	76,15
SEC	§
SBEB	overf.
SBCI	skrøbelig mænniske, som ær drek oc madek
SBCK	Suso.
SBCV	175,29
SCKF	putredo et vermis *(jf. Sir.19,3(Vulg.))*
SKFO	Lat.
SEC	§
SBEB	måske sammenblandet med dræg
SBCI	fex .. dreck eller berme .. fecula .. lyden drek vel berme
SBCK	Chr. Ped. Voc. 1510.
SBCV	63r
SEC	§
ELDK	fra
ELPS	mnt.
ELPO	dreck
ELPB	skarn etc.;
ELDK	egl. samme ord som thræk,
EEHL	jf. Bland.I.43

Figur 1: Ordbogsartikel fra GLDO for opslagsordet "dræk"

For tydeligere at vise den hierarkiske opbygning gengives ligeledes en træstruktur for artiklen "dræk".



Figur 2: Træstruktur for artiklen "dræk"

Et af de grundlæggende krav ved lagring af data er, som ovenfor nævnt, at relationerne mellem oplysningerne skal afspejles. Det vil f.eks. betyde, at sammenhængen mellem betydninger (SBEB) og tilhørende citater (SBCI) og kilder (SBCK) skal være entydig. Ved søgning i en database skal det således være muligt at få en selektiv udskrift af artiklen, omfattende f.eks. IOPO (opslagsord), IOKA (ordklasse), SBEB, SBCI og SBCK, jfr. figur 3.

IHOM	IOPO dræk
IOKA	sb.
SBEB	snavs, smuds, skarn, spec. om ekskrementer
SBCI	ther efter skulle han al thenne verldens lyst vyrde sosom drek
SBCK	Suso.
SBCI	then indwolff, som vdsender eller vd skyuder drecket ok skarnet aff mennisken
SBCK	Kv Rosg.
SBEB	overf.
SBCI	skrøbelig menniske, som ær drek oc madek
SBCK	Suso.
SBEB	måske sammenblandet med dræg
SBCI	fex .. dreck eller berme .. fecula .. lyden drek vel berme
SBCK	Chr. Ped. Voc. 1510.

Figur 3: Selektiv udskrift af artiklen "dræk"

Compulexis-artiklerne kan uden problemer overføres til DANSTATUS-poster, idet én Compulexis-artikel svarer til én DANSTATUS-post. I Vestergaard (1987) findes en detaljeret beskrivelse af overførslen. Her skal blot vises et eksempel, nemlig artiklen "dræk", figur 4.

```
IOPO        dræk
IOKA        sb.
BSUB        n.
BOKK        A089
BOKV        76,15
BFLE        sg. bek.
BOFO        -ket
SBEB        snavs, smuds, skarn, spec. om ekskrementer
SBCI        ther efter skulle han al thenne verldens lyst vyrde sosom drek
SBCK        A148
SBCV        50,8
SCKF        træk
SKFO        Sv.
SCKF        stercus
SKFO        Lat. (jf. stercora. Filip.3,8(Vulg.))
SBCI        then indwolff, som vdsender eller vd skyuder drecket ok skarnet
          aff mennisken
SBCK        A089
SBCV        76,15
SBEB        overf.
SBCI        skrøbelig mænniske, som ær drek oc madek
SBCK        A148
SBCV        175,29
SCKF        putredo et vermis (jf. Sir.19,3(Vulg.))
SKFO        Lat.
SBEB        måske sammenblandet med dræg
SBCI        fex .. dreck eller berme .. fecula .. lyden drek vel berme
SBCK        A020
SBCV        63 r
ELDK        fra
ELPS        mnt.
ELPO        dreck
ELPB        skarn etc.;
ELDK        egl. samme ord som thræk ,
EEHL        jf. Bland.I.43
```

Figur 4: Artiklen "dræk" overført til DANSTATUS

Hvis man imidlertid beder om en selektiv præsentation på skærmen, svarende til den i figur 3 viste, fås ikke det ønskede resultat, men i stedet udskriften i figur 5.

```
IOPO        dræk
IOKA        sb.
SBEB        snavs, smuds, skarn, spec. om ekskrementer
            overf.
            måske sammenblandet med dræg
SBCI        ther efter skulle han al thenne verldens lyst vyrde sosom drek
            then indwolff, som vdsender eller vd skyuder drecket ok skarnet
          aff mennisken
            skrøbelig mænniske, som ær drek oc madek
            fex .. dreck eller berme .. fecula .. lyden drek vel berme
SBCK        A148
            A089
            A148
            A020
```

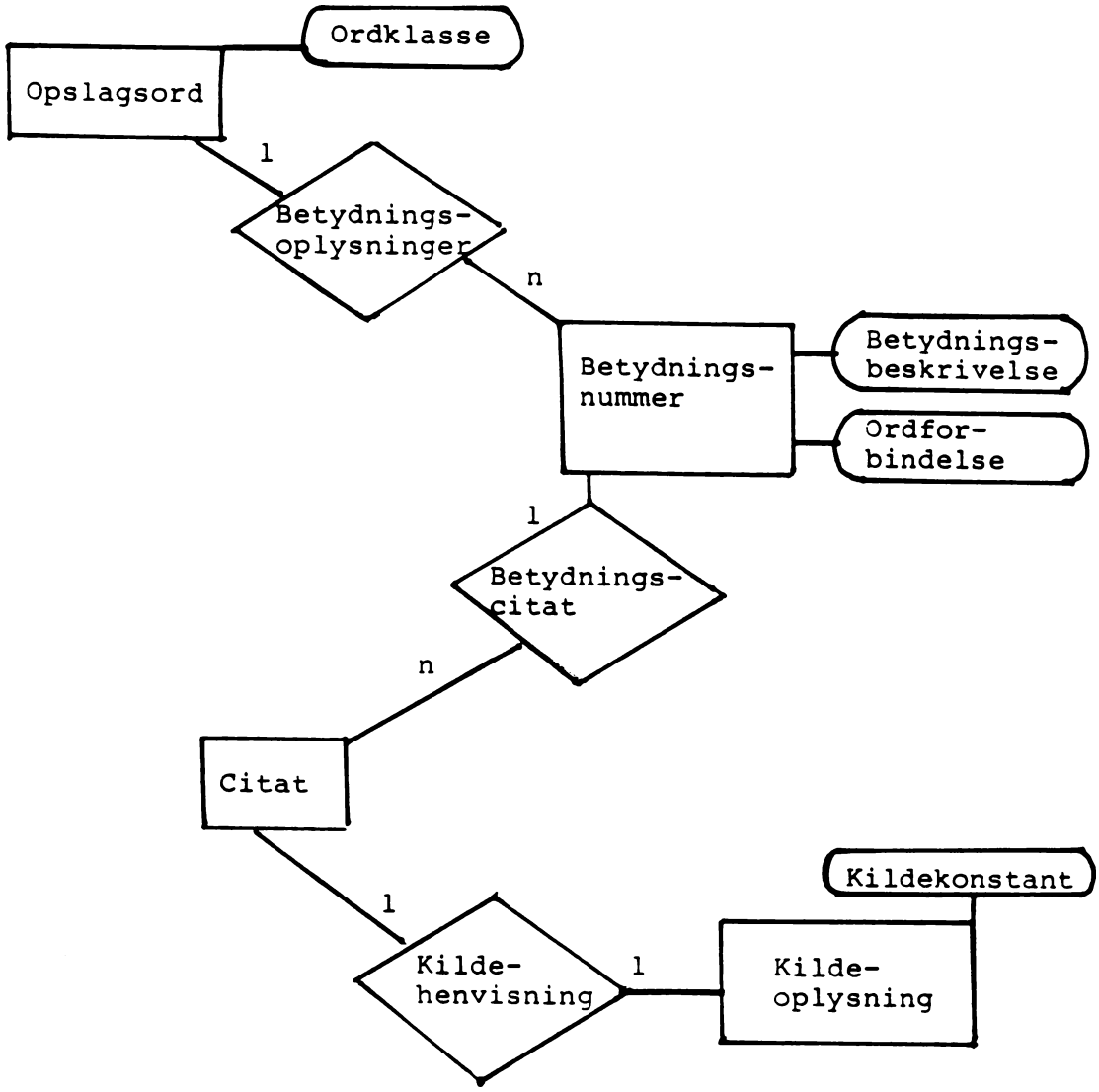
Figur 5: Selektiv udskrift af artiklen "dræk" i DANSTATUS

Denne udskrift er naturligvis utilfredsstillende, idet det ikke klart fremgår, hvilke betydninger, citater og kilder, der hører sammen. Årsagen er, at DANSTATUS opfatter alle forekomster af samme feltnavn i en artikel som ét felt fordelt på forskellige steder i posten.

3. DATASTRUKTURERING MED HENBLIK PÅ UDARBEJDELSE AF EN RELATIONSMODEL

I Ruus (1988) findes et Entitets-Relationsdiagram for S-afsnittet i GLDO. I Descriptive Tools (1987) er redegjort for hvorledes der til udarbejdelsen af et E-R-diagram for en ordbog kan anvendes både et indholds- og strukturbeskrivelsesværktøj, hhv. DANLEX-taksonomien og Warnier & Orr's klammediagram.

Med henblik på at udarbejde en relationsdatamodel, som skulle realiseres i DANSTATUS, valgtes et udsnit af S-afsnittet i GLDO, svarende til E-R-diagrammet i figur 6.



Figur 6: Udsnit af Entitets-Relationsdiagram for S-afsnittet i GLDO

Svarende til dette diagram kan der f.eks. opstilles en relationsdatamodel, som den i figur 7 viste.

I relationsdatamodellen såvel som i den efterfølgende beskrivelse er GLDO-feltnavnene udskiftet med navne, som er umiddelbart forståelige uden særligt kendskab til GLDO. Der anvendes følgende navne:

OPSL	opslagsord	= IOPO
ORDKL	ordklasseangivelse	= IOKA
BETYD	betydningsbeskrivelse	= SBEB
CITAT	betydningscitat	= SBCI
KILDE	betydningskildekonstant	= SBCK

opslagsord

OPSL	ORDKL

betydnings-
oplysninger

OPSL	BETNR	BETYD

betydnings-
CITAT

BETNR	CITAT	KILDE

Figur 7: Relationsdatamodel for udsnit af S-afsnittet i GLDO

I DANSTATUS opereres ikke med forskellige tabeller, indeholdende forskellige typer poster, sådan som det er tilfældet i et relationelt databasesystem. En DANSTATUS database består af én tekstfil og ét indeks, hvori der søges. En DANSTATUS database kan imidlertid indeholde poster med forskellig struktur, dvs. forskellige felter, og et feltnavn kan gentages inden for én post. Der afsættes ikke feltnavne eller plads til ikke udfyldte felter. Det er disse faciliteter, der er udnyttet ved forsøget med implementeringen af den relationelle datamodel.

I figur 8 er vist de tre tabeller, som er udarbejdet specielt med henblik på DANSTATUS. Tabellerne er opstillet således, at de muligheder der ligger i DANSTATUS macrofaciliteter udnyttes bedst muligt. Alle tre tabeller realiseres i én DANSTATUS database, som opbygges af tre forskellige posttyper (én for hver tabel).

opslagsord

OPSL	ORDKL	≠OPSL	≠GREN

betydnings-
beskrivelse

BETYD	≠OPSL	≠GREN

citat

CITAT	KILDE	≠OPSL	≠GREN

Figur 8: Relationstabeller for udsnit af S-afsnittet i GLDO

I tabellerne er der tilføjet nogle identifikations- eller nøglefelter:

≠OPSL	ordbogsartikelnummer (entydig identifikation af ordbogsartiklerne)
≠GREN	betydningsgrennummer (entydig identifikation af betydningerne i en ordbogsartikel)

I overensstemmelse med tabellerne i figur 8 oprettes tre posttyper i DANSTATUS, jfr. figur 9.

Posttype 1: opslagsordsposter
OPSL
ORDKL
ID
  ≠OPSL
  ≠GREN

Posttype 2: betydningsbeskrivelsesposter
BETYD
ID
  ≠OPSL
  ≠GREN

Posttype 3: betydningscitatposter
CITAT
KILDE
ID
  ≠OPSL
  ≠GREN

Figur 9: Posttyper i DANSTATUS

I figur 10 vises de 8 indlæste poster, som tilsammen udgør det valgte udsnit af den tidligere viste ordbogsartikel "dræk".

```
   1.
OPSL      dræk
ORDKL     sb
ID           #OPSL 1     #gren (1 2 3)

   2.
BETYD     snavs, smuds, skarn, spec. om ekskrementer
ID           #OPSL 1     #gren 1

   3.
CITAT     ther efter skulle han al thenne verldens lyst
          vyrde sosom drek
KILDE     Suso
ID           #OPSL 1     #gren 1

   4.
CITAT     then indwolff, som vdsender eller vd skyuder
          drecket ok skarnet aff mennisken
KILDE     Kv Rosg
ID           #OPSL 1     #gren 1

   5.
BETYD     overf.
ID           #OPSL 1     #gren 2

   6.
CITAT     skrøbelig mænniske, som ær drek og madek
KILDE     Suso
ID           #OPSL 1     #gren 2

   7.
BETYD     måske sammenblandet med dræg
ID           #OPSL 1     #gren 3

   8.
CITAT     fex .. dreck eller berme .. fecula .. lyden drek vel berme
KILDE     Chr Ped Voc 1510
ID           #OPSL 1     #gren 3
```

Figur 10: DANSTATUS-poster for artiklen "dræk"

4 SØGEMEMU BASERET PÅ MACROFACILITETERNE I DANSTATUS

Sammenkædningen af de enkelte poster, som udgør en ordbogsartikel, sker ved hjælp af søgninger på ≠OPSL og ≠GREN. Disse søgninger foregår skjult for brugeren, idet de er lagt ind i en række macroer ved hjælp af hvilke der er opbygget en særlig søgemenu.

I posttype 1, opslagsordsposterne, indføres i ≠GREN nummeret på samtlige grene i ordbogsartiklen. Herved opnås at man ikke blot kan kæde alle poster hørende til én ordbogsartikel sammen (ved hjælp af ≠OPSL), men at man også kan kæde udvalgte betydningsposter (grene) og opslagsordsposten fra én artikel sammen (ved hjælp af ≠GREN).

Menuen består af 4 hovedfaser:

(1) Søgning, herunder
- valg af søgeprofil,
- oplysning om antal svar,
- valg mellem at se
  - hele artiklen eller
  - udvalgte felter

(2) Valg mellem at se
- alle grene eller
- kun de grene hvori søgeordet findes

(3) Præsentation, herunder
- valg af profil

(4) Valg mellem at
- foretage ny søgning
- slutte

For en detaljeret gennemgang af søgemenuen og de udnyttede macrofaciliteter henvises til LAMBDA Nr. 6.

Søgemenuen bygger bl.a. på de erfaringer, der er indhøstet ved udvikling af en søgemenu til DANTERM, Dansk Termbank (Wegener 1986).

I figur 11 gengives et eksempel på en søgning ved hjælp af søgemenuen til GLDO.

```
**************************************
*                                    *
*   MENU til søgning i GLDO          *
*                                    *
**************************************
skriv søgeord
dreck
-
skriv søgefelt(er adskilt af komma)
citat

 søgeordet er fundet 1 gang i citat
 vil du se hele artikeln (y) eller nogle udvalgte felter (n)
y

1
OPSL       dræk
ORDKL      sb

BETYD      snavs, smuds, skarn, spec. om ekskrementer

CITAT      ther efter skulle han al thenne verldens lyst
           vyrde sosom drek
KILDE      Suso

CITAT      then indwolff, som vdsender eller vd skyuder
           drecket ok skarnet aff mennisken
KILDE      Kv Rosg

BETYD      overf.

CITAT      skrøbelig mænniske, som ær drek og madek
KILDE      Suso

 vil du se mere ja (y) eller nej (x)
y

BETYD      måske sammenblandet med dræg

CITAT      fex .. *dreck* eller berme .. fecula .. lyden drek vel berme
KILDE      Chr Ped Voc 1510

 vil du fortsætte søgningen (y) eller slutte (x)
y

skriv søgeord
dreck

skriv søgefelt(er adskilt af komma)
citat

 søgeordet er fundet 1 gang i citat
 vil du se hele artikeln (y) eller nogle udvalgte felter (n)
n

 vil du se alle grene (y) eller kun grene indeholdende søgeord (x)
x

 vælg præsentationsprofil (1, 2 eller 3):
1

1
OPSL       dræk
ORDKL      sb

BETYD      måske sammenblandet med dræg

CITAT      fex .. *dreck* eller berme .. fecula .. lyden drek vel berme
KILDE      Chr Ped Voc 1510
```

Figur 11: Eksempel på søgning ved hjælp af søgemenuen til GLDO

5 UDVIDELSER AF SØGEMENU

Som nævnt ovenfor er kun en del af S-afsnittet i GLDO-formatet inddraget i det beskrevne forsøg. Hvis søgemenuen skal udvides til at omfatte alle oplysningstyper i alle afsnit af GLDO-formatet, er der behov for mange nye posttyper og en betydelig udvidelse af macroerne. Såvidt det kan overskues, vil dette dog ikke medføre nye principielle problemer.

En forudsætning for at anvende metoden til et konkret projekt, er endvidere, at der udarbejdes særlige procedurer til indlæsning og ajourføring af data.

Endvidere bør menuen udvides med hjælpetekster og sikring mod forkerte svar fra brugerens side, jfr. Wegener (1986).

6 KONKLUSION

Forsøget med simulering af relationel database har vist, at der er langt bedre muligheder for datastrukturering i DANSTATUS, end den hidtidige anvendelse af systemet har tydet på.

Som tidligere nævnt anvendes DANSTATUS til en række terminologi- og ordbogsprojekter. Som eksempler kan nævnes DANTERM (Dansk Termbank) og Dansk-Fransk ordbogsbase (Blinkenberg & Høybyes Dansk-Fransk Ordbog). I begge projekter er der tale om hierarkisk strukturerede data, dog med færre niveauer end i GLDO. Der er således også her behov for en bedre afspejling af relationerne mellem data, end det er muligt at opnå ved den almindelige anvendelse af DANSTATUS, hvor én termbank- eller ordbogsartikel svarer til én post i systemet.

Forudsat at der kan udarbejdes hensigtsmæssige indlæsnings- og ajourføringsprocedurer, forekommer det derfor hensigtsmæssigt at anvende den beskrevne metode til simulering af relationel database til de to ovenfor nævnte, såvel som til andre lignende projekter.

REFERENCER

Descriptive Tools for Electronic Processing of Dictionary Data (1987), Studies in Computational Lexicography, The DANLEX Group, Danish Working Group on Computational Lexicography: Ebba Hjorth, Jane Rosenkilde Jacobsen, Bodil Nistrup Madsen, Ole Norling-Christensen, Hanne Ruus. (Lexicographica Series Maior 20), Tübingen, Niemeyer.

Nistrup Madsen, Bodil (1988): Simulering af relationel database. (LAMBDA Nr. 6), Institut for Datalingvistik, Handelshøjskolen i København.

Ruus, Hanne (1988): Lexical Data Structures. Indlæg på XIV ALLC konference i Göteborg 1987. Udkommer i Literary and Linguistic Computing 1988.

Vestergaard, Bodil (1987): Undersøgelse af databasesystemer til ordbøger. (LAMBDA Nr. 2), Institut for Datalingvistik, Handelshøjskolen i København.

Wegener, Helle (1986): Forslag til et menubaseret brugerinterface til DANSTATUS opbygget ved hjælp af systemets macrofaciliteter med henblik på søgning i HHK's termbank DANTERM, Prøve 3, Anvendt Sprogvidenskab, Linie 1: Datamatisk Lingvistik.

Krista Varantola
University of Turku

Term banks, text banks and bank users

I am going to take a very concrete, user-oriented point of view in this article. I shall restrict the user-orientation to refer to special field translators and their term bank or rather electronic data bank needs within special fields.

Background.

The Centre for Technical Terminology in Finland, TSK (Tekniikan Sanastokeskus) made its term bank, Tepa, available to the general public in the summer of 1987. The bank is directly accessible. Users, whether public institutions, private companies or private persons, sign a user contract which is free of charge. The only charge is for the use of the computer time. The rate is at present 240 FMK/h.

The bank is situated in a Helsinki University of Technology computer and is accessible 24h/day, also at weekends. The normal user prerequisites are thus a PC and a modem.

At the moment, the bank is Helsinki-centered in the sense that its use is much cheaper if you happen to live in the Helsinki area. For users outside Helsinki the telephone costs can easily become a factor prohibiting the use of the bank. In addition to normal telephone lines, it is, however, also possible to use various data transfer networks that make the telephone bill considerably smaller. I shall return to ideas of decentralisation later.

The bank is now going through its introductory phases and the number of users is still relatively low but increasing fast.

The bank consists of three data bases, Tepa1, Tepa2 and Vniiki which is a 1600 record sample of a Soviet data base consisting of GOST standards. The cyrillic alphabet has been transcribed in all data bases. The bank has now (December 1987) around 12 000 records but the numbers will double in the near future.

Tepa1 consists of a glossary on fire prevention and a glossary on labour protection. It has the highest quality rating of the

three data bases. The records come from glossaries that are the result of the co-operation between subject specialists and terminologists. In other words, the glossaries adhere to terminological principles and contain definitions and explanations that give the user an idea of the conceptual systems behind the terms.
Tepa2 is based on subject specialist work but the methodology varies and the terms often lack definitions.
The glossaries included cover such areas as welding, foundries, process automation, clothing, road and transport and forest economics.
Of the future planned data bases, TEPA3 will consist of brand new term records not yet available in printed glossaries and TEPA4 of various types of information that has been handed over to TSK.
In other words, due to pressure from the user end, various types of information are being made available but it is ultimately left to the users to assess the reliability of the information they obtain. The users' task is, however, made easier by having a well-organized bank, by using different data bases with varying reliability estimates. The bank is also interactive and users can send their comments to TSK, through the mail box of the bank.
To summarize, it could be said that the bank is not ideal but it is good enough to be useful. If we think of the slowness and costs of high quality terminology work and of user needs on the other hand, some kind of compromise must be found. It seems very sensible to make public a well-sorted small bank that still has the flexibility of reacting to user comments and changing the approach if necessary.

The project.

TSK is research-oriented and keen on co-operation with university institutions. It was thus agreed that we at the School of Translation Studies, University of Turku, should make an analysis of the term bank as seminar work and from the special field translator's angle. The preliminary work, getting acquainted with the technical aspects and defining the aims of the project has begun.

We have at our disposal a PC with a hard disk (IBM compatible), a 1200 baud modem and main frame capacity at the university computing centre. To access the term bank we do not use the normal telephone lines but the universities' data transfer network FUNET which we access through the university's mainframe DEC20. In the analysis it may also become necessary to use the university's IBM mainframe to run certain textual analysis programs. These technical aspects mean that the time taken to learn the various tricks of the computer trade is not inconsiderable. Enough time must be allocated for learning to use the equipment and utilize its potential. This is a relatively new phenomenon within the humanities but very much a fact of life that should be seen as necessary methodological training also in language and communication studies.

The aim of the project is to study the usefulness and future potential of the bank from a special field translator's point of view.
We intend to study the bank from a theoretical structural angle - analyse the instructions given to users, assess the concrete value of the various record types and use case-studies, i.e. think in terms of hypothetical translations from Finnish into a foreign language within a field present in the bank.
The software we intend to use consists, naturally, of the search routine of the term bank, the TRIP program, but also of text analysis and concordance programs available for PCs and the mainframes. With the content-analysis programs we hope to find out about the lexical and textual structure of the records in the bank and of other special field texts that we use as back-up material.

Problem-setting.
Accessing. What are the optimal ways of accessing a term bank? It is certainly a very good thing that TSK has decided to open the term bank to the public with direct access and without intermediaries. Its location in Helsinki, however, poses a problem for outside users. Telephone costs may become high in the long run if you want to do thorough searches or become bank-addicted. This is obviously not a problem for big users but

certainly for small-scale users. The computer time, on the other hand, is surprisingly cheap. (A 40 minute search which gave over 2 000 chargeable segments cost about 13 FMK).

Alternative approaches might be worth thinking of and we are therefore going to ask what kind of benefits could be derived from a multi-channel access system.

Even a human answering service might prove useful. An experienced intermediary could deal with questions coming in either electronically, by telephone or by post depending on customer needs, equipment and attitudes.

Other questions that will be asked are:

Would it make the bank more flexible if it were accessible in a number of computers in Finland ,e.g., at universities other than the university of technology in Helsinki?

Should the customers be able to buy parts that they need frequently on diskettes? Could an updating service be combined with the diskette service? Should parts of the information also be available in portable form or in a form that could be transferred into a portable electronic dictionary or handbook?

Many advantages are obvious in a multi-faceted service distributing terminology and special field text data. Future translators will rely more and more on electronic data that is directly accessible during the translation process by means of refined search routines.

A translator's work station will have (at least) desktop publishing facilities, consist of a powerful PC and a CD ROM drive, with word processing, electronic glossaries, dictionaries, enclyclopaedic information etc.

Context-oriented approaches such as REFTEX, based on syntactic and textual information available in bilingual parallel texts will certainly prove very helpful tools (cf. Kjærsgaard, forthcoming).

The worst bottleneck for individual, tailor-made data collections will be overcome when a reasonably priced scanner becomes good enough for reading in textual data from sources typeset in various ways.

Various structured special field text corpora will also be available in PC format for a number of referential purposes.(cf. e.g. Brekke, forthcoming). Commercial data banks and free

general services also provide a wealth of special language information useful for translators, on an international or national basis (e.g. the data banks accessible through Easynet, the Videotex services in Finland, France's Teletel etc.)
So, what could a central term bank's answer be to the changing concept of special field translator or communication work?
Before speculating on answers we can look at the question from the customer's point of view.

The LSP translator.

The main problems for a non-specialist LSP translator can be summarized as problem finding and problem solving. In the training we can do our best to increase the future translator's awareness of the hidden problems of LSP texts. For example, we can emphasize the rules of expert-to-expert communication, the demands of certain strictly controlled text types such as contracts and patents, the demands of manuals, instructions, text books, popularizations etc., in other words, of various types of special language communication situations (Cf. Varantola 1986 and Varantola, forthcoming).
We can also make the students aware of their lack of extralinguistic, encyclopaedic knowledge in a special field by drawing parallels from general knowledge inference patterns and associations. We can, for example, ask what kind of associations, public and private, we automatically obtain from pieces of information.
If we start a Finnish short story by saying

- It was January, smoke from the chimneys rose straight up towards the clear sky, the children's hour was about to begin. I decided to have a swim ...-,

our public associations tell us that it was cold; also, at least if you are a parent of relatively small children, that it was about six o'clock p.m. and therefore dark; the television channel in question was number two; the swim would have to take place indoors.
The private associations might include wondering whether the person involved has a private swimming pool and if so

assumptions of affluence or, on the other hand, thinking that it is the hour when the children terrorize the whole house.
If, however, the context were different, a particular technical field and the text run

- The headbox has been greatly simplified, with no internal showers, fewer rotating parts (i.e. removal of rectifier rolls and their drive)....
The sheet is essentially formed and drained at the suction breast roll, and the deflectors under the wire serve simply as support. -

our associations would be very meagre and mainly based on whatever textual cues we would obtain, unless we were experts on paper machines.
What I am driving at is that the problem-solving stage within a special field is where a central term bank can play its part and help a translator to build up the conceptual framework of the field, to cope with the associations, the information hidden in the covert inferences. (Cf. ARK 36, 1987 where the structure and contents of a multi-purpose and multi-language legal database is discussed in detail.)
We must also keep in mind that we are mainly thinking of translators who most of the time translate into a foreign language and not into their native language. It is a fact of life that can hardly be changed in the case of small languages and which adds to the types of textual information that would be helpful in special language translation.

A multi-channel, multi-base data bank

What kind of term bank, or should we say LSP data bank information, would be most useful? What is not there that should be there? How important is the quality vs. quantity distinction?

The central term bank could have, within its field of operation, the overall function of a clearing house. In this capacity it would control the information flow, the information it collects, receives or knows about, and distribute it in various electronic ways suggested above. (Copyright and resource issues are simply

disregarded in this context).
The term bank function is naturally a basic one. A bank, regularly revised and updated, is an indispensable tool in all special field communication. With this basic function under control, it is, however, possible to build some variety into it.

It has already been pointed out that multiple formats of the information would make the utilization of the data very flexible.
From the point of view of comprehension, it is the definition and explanation parts in term records that are particularly useful for translators. The definitions delimit the range of the concepts,i.e. exercise the normative power of terminologies by trying to control the indeterminacy of meaning which is such a basic quality of general language (cf. de Beaugrande, forthcoming). The explanations, commentary parts (which could be far more elaborate) also give very valuable information about the conceptual framework for the uninitiated. They relate the concept to other closely related concepts, warn of misconceptions or misleading usage, or discrepancies between the conceptual systems in different languages. This type of information builds up the insufficient extralinguistic knowledge which is most frustrating for the translator and a major source of mistranslations.
Another way of building up extralinguistic knowledge in a dictionary context is to give special field background information in the same fashion as is done in such general dictionaries as **Collins English Dictionary:**

Planck constant or **Planck's constant** n. a fundamental constant equal to the energy of any quantum of radiation divided by its frequency. It has a value of 6.6262 x 10 joule seconds. Symbol :h. See also **Dirac constant.**
There is also an entry for Max Planck:
Planck Max (Karl Ernst Ludwig). 1858-1947, German physicist who first formulated the quantum theory (1900): Nobel price for physics 1918.
This particular example was chosen because it was among the items in a quiz that was published as part of a **Newsweek**

article (April 20, 1987). The topic of the article was the type of associative knowledge literate Americans should have at high school level. What they were supposed to know about Planck's constant, according to the quiz, was that it is a quantum theory cornerstone. In other words, the associative knowledge required is not very deep, on the contrary, rather superficial but enough to keep the reader on the right track. And this is the type of knowledge a translator could also benefit from in a special field context. The background information need not always be thorough and detailed. A few helpful hints will do the job in many cases when in doubt about the correct alternative or interpretation.

Another question is, how the hints should be offered; as small-scale "cue" entries in a term bank or large scale encyclopaedic entries on a compact disk or a text bank? (See ,however, Svensén 1987:157 ff. who does not think that encyclopaedic dictionaries are very useful.)

Yet, the advent of CD ROM encyclopaedias and dictionaries, etc. does not do away with the need of more "controlled" text bases in special fields.

We can imagine that the need for terminological, i.e., lexical information and certain types of extralinguistic knowledge could be catered for by the expanding term bank, a bank that is in a constant process of updating and revision - but what would take care of the phraseological, syntactic and textual needs of the translator?

It is obvious that filling in all kinds of standard text formulas, e.g., in contracts, patents, specifications, instructions, etc. will be fast and routine if they can be found in a database.In addition, translators would also benefit from a wide selection of controlled text types and sample texts, that could be searched with sophisticated programs. In this way, translators would have at their disposal, in electronic form, parallel texts that they now mostly have to find manually (cf. however Kjærsgaard, forthcoming and the REFTEX system). Finding suitable parallel texts tends to take a great deal of time and when they are found the search procedures are not very sophisticated. With automatic search it would be possible to find differnt types of information, verb+noun, adjective+noun

collocations, use of adverbs and of prepositions, often a tricky business at least for Finns, ideas of word order, sentence structure and textual patterns. (See e.g. Picht, forthcoming). If the texts included in the base represented a wide range of text types with varying degrees of LSP-ness it would be possible to make sophisticated comparisons of term depth, strictness of term use, term density (use of fillers to control the rate of information flow), proportions of covert and overt information, grammatical features, etc. (cf. Varantola, forthcoming).
Information of this type would add to the translator's intuition of style and successful LSP communication. And, this kind of "acquired" intuition is essential in any type of textual modification task.
To speculate on the nature of the texts, it would seem sensible to include the texts from which the terms have been excerpted, but other, up-to-date texts would be welcome, too. A central LSP databank could certainly benefit from the knowledge of individual translators who, over the years, have collected valuable private term and text files that could be made more generally available through the bank. If there is no way of checking and guaranteeing the reliability of a private or a company collection, it should still be made available with the necessary reservations, because top quality terminology work is extremely time-consuming. I do not think there is any need to hold back information if its expected status is clearly indicated.
In other words, quality is the key concept but the need for quantity and versatility is the pragmatic demand that has to be satisfied. And, there seems to be little reason why the two seemingly opposite demands could not be combined. Basic research into text selection and ideal term record structure is essential. New findings will in due course be applied in practice but, at the same time, an existing bank has to react to customer demands. If the bank is well-structured with clearly differentiated data banks and not a Christmas pudding type mixture, it should be possible to maintain it in a constant stage of revision and development. Up-to-dateness and flexibility are after all the basic advantages of an electronic data bank.

It seems, however, that the term and text bases should remain separate but mutually compatible, e.g., in the sense that you would know that each term present in the term bank would also be available in a text context.

A clearing house of the type sketched above would naturally involve an immense amount of work. On the other hand, there is no reason why it should be done on a small scale. Co-operation between LSP data bases is necessary,as is exchange of material and the use of other data bases compiled for other than LSP term- or textbank purposes. General services of the French leletel type and others mentioned above, could prove selectively valuable.

It is also clear that although I have taken a narrow LSP translator's view, other groups of users, subject specialists naturally, but also industry in general, professional training, LSP research, etc. would benefit from a multifaceted system and could also contribute to it.

In this article I have tried to bring up ideas that we hope to study empirically during our small-scale project. It is an advantage that we have a chance of this hands-on approach at this stage when the bank is still small and easier to conceptualize than a large bank.

REFERENCES

ARK 36. 1987. Sproginstitutternes Arbejdspapir. Handelshøjskolen i København. Copenhagen.

de Beaugrande, R. Forthcoming. "Text as the new foundation for linguistics". Paper presented at the VIth European Symposium in LSP, Vaasa 1987.

Brekke M. Forthcoming. "The Bergen English for science and technology (Best) corpus: A pilot study". Paper presented at the VIth European Symposium in LSP, Vaasa 1987.

Collins Dictionary of English. 1979. Glasgow.

Kjærsgaard, P.S. Forthcoming. "REFTEX - A context-based translation aid". Paper presented at the 3rd Conference of ACL European Chapter, Copenhagen 1987.

Newsweek. 1987.(April 20). "A Dunce Cap for America. What we don't know, why we don't know it".44-46.

Picht, H. Forthcoming. "Fachsprachliche Phraseologie". Paper presented at the VIth European Symposium in LSP, Vaasa 1987.

Svensén, B. 1987. Handbok i lexikografi. Stockholm

Varantola, K. 1986. "Special language and general language". Unesco Alsed-LSP Newsletter, 9:2. 233-244.

Varantola, K. Forthcoming. "LSP translation from a linguistic angle with special reference to engineering texts". Paper presented at a symposium on translation and LSP , University of Surrey 1986.

Varantola K. 1987. "Popularization strategies and text functional shifts in scientific/technical writing". To appear in UNESCO-Alsed LSP Newsletter.

JURAPROJEKTET

ERFARINGER OG RESULTATER

Inge Gorm Hansen, Institut for Engelsk, HHK.

Juraprojektet blev iværksat ved HHK i efteråret 1984 med pilotprojektet: Database til terminologisk information og generering af ordbøger. På symposiet i 1985 i Helsinki gav Steffen Leo Hansen en statusrapport over projektet, som er offentliggjort i symposiets proceedings [1]. Projektgruppen [*] har i begyndelsen af 1987 afsluttet projektet og udarbejdet en større rapport (ca. 500 sider incl. bilag) til Egmontfonden, som har støttet projektet økonomisk. En oversigtsartikel om pilotprojektet er udkommet i HHKs publikation ARK no. 36 [2]. I det følgende vil projektets enkelte faser således ikke blive gennemgået i detailler, idet der henvises til ovennævnte publikationer.

Målet med dette indlæg har i højere grad været at fremlægge nogle konkrete arbejdsresultater og perspektivere dem i forhold til fremtidig terminologisk og leksikografisk forskning.

Oversigt over pilotprojektet

Formål: at udvikle og afprøve en korpusbaseret arbejdsmetode, der forener terminologiske og leksikografiske principper, og som skal resultere i

- en flersproget terminologi-database
- sprogretnings- og brugerbestemte tosprogs-ordbøger
- den fagsproglige komponent af en større etsprogs-ordbog for danske brugere

[*] Steffen Leo Hansen, Ole Helmersen, Bodil Nistrup Madsen, Hanne Puggård, Joan Haff Tournay, Charlotte Werther, Anne Zoëga og Inge Gorm Hansen

Sprog: dansk, engelsk, spansk

Emne: civilproces - borgerlige domssager i 1. instans

Baggrund:
- manglen på egnede, fagspecifikke ordbøger for danske brugere
- ønsket om at kombinere anvendelsen af EDB med nyere forskningsresultater inden for terminologi og leksikografi ved fremstillingen af ordbøger

Målgruppe: danske brugere
- oversættere og tolke
- jurister
- advokatsekretærer
- undervisere i juridisk fagsprog
- jurastuderende
- erhvervssproglige studerende

Arbejdsfaser:
- oprettelse og anvendelse af tekstkorpus
- inventarisering af fagsproglige termer
- oprettelse af en flersproget terminologi-database
- generering af ordbøger

Emnet for pilotprojektet er som omtalt fagsprogligt. Jura er valgt som fagområde, da der inden for dette oversættes og tolkes meget, og da der generelt savnes egnede juridiske ordbøger. Samtidig udgør emnet et område, hvor man må forvente at blive præsenteret for en lang række metodiske og indholdsmæssige problemer, når det skal bearbejdes terminologisk og leksikografisk. Især ækvivalensproblematikken er et meget vanskeligt område indenfor juridisk sprog, hvor man arbejder med vidt forskellige referencerammer i form af de forskellige juridiske systemer.

Arbejdsfasen, oprettelse af tekstkorpus, har været projektets mest omfattende og har bestået af

1. etablering af tekstkorpus
2. fastlæggelse af korpusstruktur
3. oprettelse af et maskinlæsbart korpus
4. udvikling af programmel til automatisk korpusanalyse

Pilotprojektets to sidste arbejdsfaser, oprettelse af en flersproget terminologidatabase og generering af ordbøger, har især omhandlet anvendelse af DANTERM-recorden i forbindelse med oprettelse af arbejdsdatabasen JURTERM.

Anvendelse af det maskinlæsbare korpus

I det følgende vil anvendelsen af det maskinlæsbare korpus, blive uddybet med eksempler på søgning efter data i korpus til indlæsning i terminologidatabasen. Der vises eksempler på, hvorledes data genereres fra terminologidatabasen til ordbogsprofiler beregnet for forskellige brugergrupper.

Pilotprojektets korpus består af et "manuelt" og et maskinlæsbart korpus, hvor det sidste er en delmængde af det første. Det manuelle korpus er emnedækkende for civilproces og udarbejdet i form af en bibliografi i samarbejde med juridisk laboratorium ved Københavns Universitet. Det har ikke været muligt indenfor projektets rammer at etablere det totale korpus i maskinlæsbar form. Teksterne i det maskinlæsbare korpus er således primært valgt ud fra de specielle krav, man må stille til terminologisk dokumentationsmateriale og på baggrund af teksttypologiske overvejelser og spørgsmålet om repræsentativitet.

Det samlede korpus - dansk, engelsk, spansk - består af ca. 1 million løbende ord og indeholder følgende teksttyper: love, lovkommentarer, lærebogstekster, retsafgørelser og partsskrifter (formularer og autentiske dokumenter).

Hvis man sammenligner jurkorpus med andre korpora, kan det forekomme meget omfattende, når man tager i betragtning, at det dækker et ret begrænset emneområde indenfor et enkelt af juraens fagområder. Dette skal ses på baggrund af føromtalte specielle krav til korpus, som projektgruppen har formuleret. Det har været et væsentligt krav til korpus, at det såvel kvalitativt som kvantitativt skulle være dækkende for alle inden for det udvalgte emneområde forekommende teksttyper. Korpus skal ikke blot kunne anvendes til at registrere sprogbrugen i de forskellige teksttyper, men også fungere som en videns/dokumentationsbase, hvor man kan hente faktuelle oplysninger og søge efter definitioner af termer. Målet har således været, at korpus ved sin sammensætning kunne tilgodese ønsket om at registrere både lingvistiske og ekstralingvistiske faktorer i forbindelse med de enkelte termer.

Dette skal ses i sammenhæng med den forskel, der traditionelt er mellem leksikografiske og terminologiske arbejdsmetoder. Leksikografen vil normalt tage det enkelte ord som sit udgangspunkt og beskrive hele ordets betydningsfelt, hvor terminologen isolerer én betydning i det samlede betydningsfelt som led i en systematisk beskrivelse af et begreb, dets relationer til over- og underbegreber etc. Det foreliggende projekt er en blanding af de to nævnte arbejdsmetoder, hvorfor der stilles specielle krav til sammensætningen af projektets tekstkorpus.

I den konkrete anvendelse af det maskinlæsbare korpus har der været gennemført en række forsøg med søgning efter data med ovennævnte to mål for øje: 1) registrering af sprogbrug og 2) søgning med henblik på at finde definitorisk materiale til indholdsbeskrivelse af de enkelte termer.

I forbindelse med 2) har der været gennemført forskellige forsøg med strategier i on-line søgning. Teksterne er klassificeret med nøgleord og ved hjælp af nøgleordskoderne har man forsøgt at fin-

de frem til tekststeder med faktuelle oplysninger, som kunne bidrage til definition af termerne. Andre forsøg er gået ud på at søge på forskellige "strenge". Man har f.eks. kunnet søge på "stævning + forstås" for at identificere et tekstafsnit indeholdende ordene "ved stævning forstås", eller f.eks. "stævning ...er". Det har givet resultater i visse tilfælde, men er et område, som bør undersøges nøjere, da de forsøg der har været gennemført endnu hviler på et spinkelt empirisk materiale. Der er således i forbindelse med eksemplet "stævning" valgt en definition, som ligger udenfor det maskinlæsbare korpus som den overordnede definition samt en udvidet definition, som er trukket fra det maskinlæsbare korpus.

Med hensyn til 1) sprogbrugsundersøgelserne, registrering af fraseologi, kollokationer etc. er der udarbejdet en række programmer, som er beskrevet i projektgruppens rapport. Steffen Leo Hansen har i forbindelse med projektrapporten udarbejdet Jurkorpus 1 og 2, hvori er vist eksempler på konkordansprogrammer m.v. Det første nummer af Inst. f. Datalingvistiks publikation LAMBDA [3] indeholder instituttets programbibliotek, hvor jurkorpusprogrammerne er dokumenteret.

Her skal blot omtales to konkordansprogrammer, som forfatteren har arbejdet med. Det drejer sig om et liniekonkordansprogram (KWIC-index) og et sætningskonkordansprogram. I liniekonkordansprogrammet er der mulighed for ved søgning på en bestemt term at opstille en liste med alle belægssteder, hvor den pågældende term f.eks. er midtstillet eller står først eller sidst på linien. Der vises således blot de ord, som omgiver den pågældende term begrænset til én linie. Med sætningskonkordansprogrammet fremkommer ved søgning på f.eks. ordet stævning alle de sætninger i korpus, som indeholder ordet stævning. Søgeordet kan "highlightes". I forbindelse med udarbejdelsen af recorden for "stævning", som vises senere, har forfatteren gennemført søgninger efter kollokationer m.v. både ved hjælp af KWIC-indexet og sætningskonkordansen. Søgningerne er gennemført og resultaterne registreret for

hver enkelt af teksttyperne i korpus med henblik på at sammenligne sprogbrugen i de forskellige teksttyper og resultaterne fra de to konkordansprogrammer. En overraskende høj procentdel af de registrerede kollokationer kunne "fanges" ved gennemgang af KWIC-indexet, som er meget overskueligt og hurtigt at arbejde med. Sætningskonkordanserne var naturligvis betydeligt mere omfattende og mere tidkrævende at arbejde med. En sætningskonkordans for ordet stævning havde således et omfang af 60 sider og KWIC-indexet blot 10, hver dækkende 418 belægssteder i den danske del af korpus.

Det er forfatterens opfattelse, at de muligheder for registrering og sammenligning af sprogbrug i forskellige teksttyper på dansk og det/de relevante fremmedsprog, som et maskinlæsbart korpus giver, vil kunne forbedre kvaliteten i fagsprogsordbøger betydeligt. Ved anvendelse af et tekstkorpus vil man kunne få svar på en lang række af de problemer, som gængse ordbøger ikke giver løsningen på i dag. Der tænkes her på de mange forbindelser, i hvilke et ord optræder i forskellige tekster, hvilke præpositioner, verber etc. de enkelte termer kan forbindes med, aspekter, som sjældent belyses i fagordbøger. Den metode, som oversættere af højt specialiserede fagsproglige tekster anvender til løsning af denne type sproglige problemer, er normalt at arbejde med originale kilder omhandlende det pågældende emne og ved læsning af disse indsamle oplysninger om sprogbrugen. Denne metode er meget tidrøvende og ville kunne lettes betydeligt ved anvendelse af maskinlæsbare korpora.

Anvendelse af arbejdsdatabasen JURTERM

Efter udvælgelse af teksteksempler, definitioner m.m. fra det maskinlæsbare korpus lægges disse oplysninger over i en fil, og de pågældende data kan overføres maskinelt til databasens records, (jf. bilag 1). Oplysninger, som hentes fra kilder udenfor det maskinlæsbare korpus, indtastes. I første omgang udfyldes records

med danske oplysninger ved hjælp af en "maske" med på forhånd indsatte feltnavne, dernæst udfyldes for hver record engelske, spanske etc. "delmasker", som flettes sammen med de danske oplysninger.

Projektgruppens arbejde med DANTERM-recorden har bevirket, at den er blevet udvidet og ændret med henblik på at kunne tilgodese de krav, der er opstået i forbindelse med juraprojektet.

DANTERM-recorden er meget omfattende, da den er opbygget med henblik på at kunne rumme alle de oplysninger, som en række forskellige brugere måtte have behov for at registrere. I den enkelte record kan man således begrænse sig til kun at udfylde de kategorier, som er relevante for et givet projekts målgruppe.

Som nævnt har pilotprojektets målgruppe været oversættere og tolke, jurister, advokatsekretærer, undervisere i juridisk fagsprog, jurastuderende og sprogstuderende. Med henblik på at gennemføre forsøg med forskellige typer af ordbogsartikler/profiler til de nævnte målgrupper har det været nødvendigt at udfylde de fleste af recordens felter. Ved symposiet udleveredes to eksempler på udfyldte records for "stævning" i terminologidatabasen, som skulle illustrere anvendelsen af de enkelte kategorier til registrering af oplysninger af både sproglig og ekspertfaglig art. Det drejer sig om en record indeholdende dansk, engelsk og spansk og en record indeholdende engelsk. De udfyldte records er meget omfattende især på grund af de faktuelle oplysninger, hvor mange sider tekst er registreret og f.eks. stævningsformularer er indlæst. Af pladshensyn er de udfyldte records ikke medtaget her.

Eksempler på brugerprofiler

Som nævnt skulle de udfyldte records dække alle de definerede brugergruppers behov, og i ovennævnte records for termen stævning er de fleste felter udfyldt. På basis af oplysningerne i recorden kan der således sammensættes forskellige sæt af oplysninger -

brugerprofiler - til de enkelte målgrupper. Hver profil udgør en delmængde af oplysningerne i den samlede record. Der er endnu ikke foretaget forsøg med automatisk generering af sprogretnings- og brugerbestemte ordbøger, da der på nuværende tidspunkt ikke er indlæst et tilstrækkeligt stort antal records i databasen. Automatiseringen af genereringsprocessen vil bl.a. bestå i anvendelse af på forhånd definerede profiler. For at illustrere, hvordan sådanne profiler kan sammensættes, er der udarbejdet eksempler herpå.

Ved symposiet udleveredes følgende eksempler på brugerprofiler:

1) Profil sammensat med henblik på en jurastuderende

2) Profil sammensat med henblik på en oversætter - sprogretning dansk-engelsk.

3) Profil sammensat med henblik på en jurist - sprogretning dansk-engelsk.

Alle eksempler er udarbejdet for termen stævning.

Her skal af pladshensyn kun vises 1) og 2), da 3) er meget omfattende. Feltet FACTS indeholder således 8 sider tekst.

Nedenfor vises 1), som er en dansk record til brug for f.eks. en jurastuderende.

Som det ses, drejer det sig her om en profil sammensat udelukkende på basis af de danske oplysninger i recorden. Den danske del af terminologidatabasen vil således kunne anvendes i forbindelse med uarbejdelse af en et-sprogs systematisk definitions- og sprogbrugsordbog. Det skal understreges, at den profil, der er sammensat for en jurastuderende kun er et eksempel og ikke en fastlagt profil. Den studerende kan have behov for at få defineret et begreb, og få oplysninger om over- og underbegreber. Der kan ligeledes være behov for informationer om sprogbrug, når den studerende skal skrive opgaver, udfærdige eksempler op proces-skrifter m.v.

EKSEMPEL 1

1.

DA_TERM stævning

DA_DEF 1 stævning: sagsøgerens første processkrift, der skal indeholde påstand, en kort fremstilling af de kendsgerninger, hvorpå påstanden støttes, og angivelse af de beviser som påberåbes samt parternes navne og adresser og retten, hvor sagen anlægges, jfr. Rpl. § 348.
(kilde: vEyJ - 5 s. 212)

2 Stævningen skal indeholde:

1) parternes navn og adresse, herunder angivelse af en postadresse i Danmark, hvortil meddelelser til sagsøgeren vedrørende sagen kan sendes, og hvor forkyndelse kan ske,
2) angivelse af den ret, ved hvilken sagen anlægges,
3) sagsøgerens påstand,
4) en kort fremstilling af de kendsgerninger, hvorpå påstanden støttes, og
5) angivelse af de dokumenter og andre beviser, som sagsøgeren agter at påberåbe sig.

(kilde: Rpl § 348 stk. 2)

DA_EXT Stævningen skal være udfærdiget i A 4-format og indleveres til den kompetente ret med 3 genparter, samt en genpart af de dokumenter, som sagsøgeren agter at påberåbe sig, for så vidt de er i hans besiddelse, jfr. § 348, stk. 2 og 3.
(kilde: GomC, s. 63)

DA_SYNTAG affatte stævning, (kilde: GomC s.65-70)
støtte afgørelse på stævning , (kilde: GomC s.72)
anføre i stævning , (kilde: DRSH s.1, KarR s.3117, DRL s.2)
angive i stævning , (kilde: GomC s.72-78, KarR s.3139-3140, Rpl § 354)
bekendt; gøre sagsøgte bekendt med stævning, (kilde: GomC s.108-111)
beramme en stævning til forberedelse, (kilde: GomC s.205-210)
berigtige stævning, (kilde: KarR s.3143-45)
dom efter stævning , (kilde: DSGDFU)
forhåndskontrol af stævning , (kilde: GomC s.72-78, note 23)
forhåndsprøvelse af stævning , (kilde: KarR s.3141)
forkynde stævning for én , (kilde: GomC s.63-65)
forkynde stævning for én på bopælen ved en anden, (kilde: KarR s.3100)
forkyndelse af stævning , (kilde: KarR s.3100, GomC s.63-65)
forsyne stævning med påtegning , (kilde: KarR s. 3139-40)
fremlægge stævning, (kilde: GomC s.111-113, DSSK1)
genparter af stævning , (kilde: KarR s.3139-40, DVL s.2)
i; stævning i en sag, (kilde: DRSH, s.1)
imødegå stævning , (kilde: GomC s.247-249)
indgive stævning, (kilde: KarR s.3151-52)
indlevere stævning til berammelse, (kilde: KarR s.3114)
indlevere stævning til retten, (kilde: GomC s.63-65, Rpl § 251, KarR s.3119-20)
indlevering af stævning , (kilde: GomC s.63-65, KarR s.3121, Rpl § 354)
krav til stævning , (kilde: KarR s.3152)
lyder; stævning der lyder på, (kilde: DRSH s.1)
mangelfuld stævning, (kilde: KarR s.3141)
mangler ved stævning , (kilde: KarR 3136, GomC s.111-113)
meddele en stævning berammelsespåtegning, (kilde: KarR s.3157)
mod; stævning mod sagsøgte, (kilde: KarR s.3139-40)

modtage stævning i sag, (kilde: GomC s.148-150)
modtagelse af stævning, (kilde: GomC s.550-555)
opregne i stævning , (kilde: GomC s.72-78)
prøvelse af stævning , (kilde: KarR s.3141)
påberåbe i stævning , (kilde: DRSH s.1, KarR s.3100, GomC s.557-560)
påstand; stævning med påstand om, (kilde: DRSH s.1)
påstå; stævning, hvorved en sagsøgt påstås tilpligtet, (kilde: KarR s.3139-40)
påtegning på stævning , (kilde: Rpl § 360, DSUR2 s.1)
reparere mangelfuld stævning , (kilde: KarR s.3136)
rette stævning imod én, (kilde: GomC s.540)
tage stævning til følge, (kilde: GomC s.72-78)
tilbagekalde stævning , (kilde: KarR s.3141)
udarbejdelse af en ny stævning, (kilde: GomC s.159-161)
udformning af stævning, (kilde: KarR s.3069-3395, GomC s.114-119)
udfærdige stævning, (kilde: KarR s.3139-40, Rpl § 354, GomC s.63-65)
udfærdigelse af stævning, (kilde: KarR s.3139-40)
udtage stævning i sag, (kilde: KarR s.3124-25)
udtage stævning mod én eller for én ved værgen, (kilde: KarR s.3125-26,
DANSUR2 s.2)
udtagelse af stævning, (kilde: GomC s.412-413, KarR s.3133-34, DSGDFU s.22)
underskrive stævning, (kilde: KarR s.3125, GomC s.270-274)

DA_SYST sagens forberedelse
DA_POS 2
DA_REL

BC-GEN	processkrift
BC-PART	skriftveksling
NC-GEN	byretsstævning
	landsretsstævning
	adcitationsstævning
	interventionsstævning
NC-PART	påstanden i stævningen
	krav i stævningen
	sagsfremstillingen i stævningen
	sagsbehandlingen i stævningen
	stævningens fremstilling af kendsgerninger
	bevisangivelsen i stævningen
	bevisfortegnelsen i stævningen
	stævningens dokumentfortegnelse
	stævningens bevisfortegnelse
	stævningens beløb
	stævningsbeløb
	bestævnet beløb
	påstævnet beløb
PREC	udtagelse af stævning; skriftveksling
SUCC	indlevering af stævning til retten; svarskrift
CAUS	søgsmålsgrund
RC	ankestævning
	at stævne
	at bestævne
	at indstævne
	at påstævne
	stævnevarsel
	stævningsgebyr
	stævningsmand
	stævningsmandsforkyndelse

Nedenfor vises den anden profil, som er sammensat med henblik på en oversætter:

1. EKSEMPEL 2

DA_TERM stævning
DA_GRAM sub -en -er
DA_STYL lov, lovkommentar, lærebog, processkrift, retsafgørelser, retsbogstilførsler
DA_SYNTAGEN
forkynde en stævning for én (kilde: GomC s.63-65) [1]
forkyndelse af stævning (kilde: KarR s.3100, s.3142-43) [2]
forsyne stævning med påtegning (kilde: KarR 3139-40) [3]
fremlægge stævning (kilde: GomC s.111-113, DSSK1) [4]
genparter af stævning (kilde: KarR s.3139-40, DVL s.2) [5]
i; stævning i en sag (kilde: DRSH s.1) [6]
lyder; stævning der lyder på (kilde: DRSH s.1) [7]
påstand; stævning med påstand om (kilde: DRSH s.1) [8]
påtegning på stævning (kilde: Rpl § 360, DSUR2 s.1) [9]
reparere mangelfuld stævning (kilde: KarR s.3136) [10]
udtage stævning (kilde: KarR s.3124-25, DANSUR2 s.2) [11]
udtagelse af stævning (kilde: KarR s.3133-34, GomC s.65-70 DSGDFU s.22) [12]
EN_TERM writ of summons
EN_ALT ABBR writ
EN_GRAM sub -s
EN_STYL lovstof, lærebog, processkrift
EN_SYNTAGDA

amend; to amend the writ (kilde: RSC O.13,r.7) [10]
copy of the writ (kilde: RSC O.13,r.7) [5]
indorse; to indorse a writ (kilde: OHAC s. 194) [3]
indorsement of the writ (kilde: RSC O.18,r.15) [9]
issue; to issue a writ (kilde: RSC O.13,r.6) [11]
the issue of the writ (kilde: RSC O.29,r.1) [12]
produce; produce a writ (kilde: RSC O.42,r.5) [4]
serve; to serve a writ on sby (kilde: OHAC s. 206) [1]
service of the writ (kilde: OHAC s. 139) [2]
a writ claiming ... (kilde: OHAC p194) [7 + 8]
writ; the writ in the action (kilde: RSC O.18,r.6) [6]

EN_NOTE 'writ' og 'writ of summons' er alternative former og den forkortede form 'writ' er i ovenstående tekst- og syntagmeeksemple: anvendt i henhold til almindelig praksis, jfr. nedenfor.
Rules of the Supreme Court Practice 1985 indeholder i O.1,r.4(1) en række definitioner, bl.a. følgende:

> "In these rules, unless the context otherwise requires, the following expressions have the meanings hereby respectively assigned to them, namely - ... "writ" means a writ of summons"

EN_EQDA Den danske term stævning og den engelske term writ (of summons) er ikke ækvivalente, da de to retssystemer er fundamentalt forskellig.
I engelsk ret findes en række forskellige typer stævninger alt efter sagens genstand og ved hvilken domstol sagen anlægges, j.fr. beslægtede begreber anført under RC i recorden.
På basis af en sammenligning af de anførte definitioner og det under EXPL. anførte skønnes det hensigtsmæssigt at anvende writ (of summons) til oversættelse af stævning.
Endelig skal nævnes at dette iflg. gængse ordbøger er alm. oversættelsespraksis.

I ovenstående eksempel er en del oplysninger sorteret fra i forhold til oplysningerne i den første profil. F. eks. er definitionerne ikke medtaget, og der er kun anført "sprogparrelaterede kollokationer. Selvom oversætteren først og fremmest har brug for at få at vide, hvad en term på udgangssproget kan oversættes med på målsproget, vil det normalt være nødvendigt inden for fagsprog at anføre en definition eller forklaring, da oversætteren ikke kan have fagindsigt i de mange områder han/hun præsenteres for i oversættelsesøjemed. Det skal igen understreges, at de viste ordbogsprofiler kun er eksempler. Det er ligeledes nødvendigt med oplysninger om ækvivalens, især indenfor et område som juridisk sprog, hvor retsstystemer og retsvirkninger adskiller sig væsentligt fra land til land, og ækvivalensen derfor ofte kun er tilsyneladende.

Som nævnt ovenfor har projektgruppen ønsket at kunne registrere den type oplysninger om sprogbrug, som oversætteren for det meste savner i traditionelle fagsprogsordbøger. Det drejer sig om kollokationer m.v., som anført i feltet SYNTAG. Det er her muligt at kvalificere visse oplysninger, som udelukkende er relevante for ét sprogpar, f.eks. dansk-engelsk og ikke dansk-spansk. Feltnavnet DA-SYNTAGEN indeholder således "syntagmer", der er parallelle med engelske "syntagmer". I eksemplet er kollokationerne i det danske og det engelske felt anført hver for sig i alfabetisk rækkefølge. I en egentlig ordbogsartikel bør informationerne naturligvis flettes, således at parallelle danske og engelske tekstek-sempler står ved siden af hinanden. Det har imidlertid været vanskeligt at finde en metode til klassifikation af oplysningerne i feltet SYNTAG. Yderligere undersøgelser er nødvendige for at dette problem kan løses tilfredsstillende. Endvidere skal der arbejdes med metoder til automatisk sammenføring af parallelle kollokationer. I eksemplerne er relationerne mellem sprogparrene markeret med tal i kantet parentes.

Konklusioner og generelle bemærkninger

Ovenfor er de krav til korpussammensætning, som projektgruppen har stillet, kort beskrevet. Efter forfatterens opfattelse har

disse krav været for ambitiøse, da det kræver uforholdsmæssigt store tekstmængder, når man skal tilgodese et ønske om, at korpus skal kunne anvendes til sprogbrugsundersøgelser og samtidig fungere som en aktuel vidensbase. Såfremt en jurist skulle kunne "slå op" i korpus med henblik på at se, hvad der er gældende ret på et bestemt område, vil dette kræve et korpus, som kan fungere som et opdateret juridisk dokumentationssystem. Dette kunne måske gennemføres for et meget begrænset juridisk emne, men ville være urealistisk, hvis større fagområder skulle inddrages. Det ville være mere hensigtsmæssigt til vort formål at dele korpus i to, således at det ene tilgodeså sprogbrugssiden og det andet dokumentationssiden, hvor man måtte overlade vidensbaseopbygningen til andre og gennem samarbejde trække på disse vidensbaser. Via netværk er det allerede muligt at benytte forskellige juridiske dokumentationsbaser, og der arbejdes i flere lande med udvikling af juridiske ekspertsystemer. Det vil næppe være økonomisk muligt at opbygge og vedligeholde sådanne systemer alene til brug for ordbogsproduktion, selv med den nyeste OCR-teknologi.

Et andet problem i forbindelse med udnyttelse af tekstkorpus har været de meget store datamængder, som skulle gennemgås ved registrering af sprogbrug. Som nævnt fandtes 418 belægssteder for termen "stævning" alene i det danske korpus. Selv med systemets muligheder for at søge på delmængder i korpus virker on-line søgning tung, da man skal "blade" mange eksempler igennem og læse dem på skærmen. Det vil i sådanne tilfælde ofte være at foretrække at arbejde med udskrifter f.eks. i form af konkordanser etc.

Med henblik på at lette overførslen af data fra korpus til terminologidatabase har projektgruppen ønsket mulighed for at kunne se dele af korpus og dele af terminoligidatabasen samtidig. Forbedrede metoder - f.eks. "vinduer" med interaktion mellem disse - ville gøre arbejdet med udfyldning af records nemmere.

Selvom det anvendte edb-system (DANSTATUS) ikke umiddelbart er tilstrækkeligt fleksibelt som arbejdsredskab i forbindelse med overførsel af data fra korpus til terminologidatabasen, viser erfaringerne fra pilotprojektet generelt, at den edb-støttede arbejdsmetode med on-line søgning og anvendelse af udskrifter i form af f.eks. konkordanser er et værdifuldt supplement, som letter og forbedrer arbejdet i sammenligning med traditionelle arbejdsmetoder. F.eks. kan udvælgelse af konteksteksempler, genfinding af belægssteder og teksttypologiske undersøgelser gennemføres langt hurtigere og med større pålidelighed.

Som tidligere nævnt har et af målene med projektet været at forsøge at forene terminologiske og leksikografiske arbejdsmetoder og principper. Indenfor leksikografien forskes bl.a. i ordbogtypologi og artikelstruktur, områder som er særdeles relevante for pilotprojektet. Brugeraspektet i forbindelse med ordbogsproduktion er således et fælles problem for leksikografer og terminologer. Der arbejdes med receptions- og produktionsordbøger eller passive og aktive ordbøger, hvor man skelner mellem brugere med og uden modersmålskompetence og indretter kravene til artikelstruktur og indhold derefter. Målet er bl.a. at "spare" en række oplysninger f.eks. af encyclopædisk art, hvor disse oplysninger forudsættes at være brugeren bekendt.

Spørgsmålet er, om disse teorier og metoder umiddelbart kan anvendes i forbindelse med fagsproglig leksikografi og terminologi. Det er forfatterens opfattelse, at der på dette punkt må skelnes mellem fagsprog og ikke-fagsprog, da f.eks. "modersmålsbrugeren" vil have en betydeligt mindre sproglig kompetence og begrænset viden om faktiske forhold, når det drejer sig om fagsprog. Det ville være ønskeligt med en fælles forskningsindsats indenfor terminologi og leksikografi med fokus på brugerbehov generelt.

Også inden for området oversættelsesteori synes der at være behov for en øget forskningsindsats med henblik på fagsprog og fagsproglig oversættelsesstrategi. Her er det især ækvivalensproblematikken, som inden for et område som oversættelse af juridiske tekster volder store vanskeligheder.

Afslutningsvis skal de videre perspektiver for juraprojektet omtales. Som det er fremgået af ovenstående, bør der arbejdes videre med flere af projektets faser, især de to sidste faser, oprettelse af en terminologidatabase og generering af ordbøger. Det videre arbejde vil være koncentreret om udbygningen af terminologidatabasen og udvikling af metoder til automatisk generering af sprogretnings- og brugerbestemte ordbøger. Det vil være ønskeligt, om sprogene fransk og tysk, som oprindeligt planlagt inddrages i projektet. Hvorvidt projektet kan videreføres afhænger af, om der kan tilvejebringes de tilstrækkelige menneskelige og økonomiske ressourcer. Det kan i den forbindelse oplyses, at medlemmer af projektgruppen forhandler med et forlag, som er interesseret i at udgive en dansk-engelsk juridisk ordbog i samarbejde med det erhvervssproglige fakultet.

Referencer:

1) Symposium om Datorstö Terminologi och Lexikografi, Helsingfors 1985, s. 29
2) Ark 36, HHK, 1987
3) Lambda 1, Institut for Datalingvistik, HHK 1986

DELTAGERLISTE

NORDISKE DATALINGVISTIKDAGE

3.-4.11 1987

FINLAND

Centralen för teknisk terminologi
Elisabetsgatan 16 D
SF-00170 Helsingfors

Timo Honkela
SITRA Foundation
P.O. Box 329
SF-00121 Helsingfors

Tuomo Tuomi
Forskningscentralen för de inhemska språken
Elisabetsgatan 16 A 1
SF-00170 Helsingfors

Krista Varantola
Department of English
University of Turku
SF-20500 Turku

NORGE

Tove Fjeldvig
Statens Datacentral
Ulveveien 89 B
N-0581 Oslo 5

Anne Golden
Inst. f. norsk som fremmedspråk
Universitetet i Oslo
Postboks 1066
Blindern, 0316 Oslo 3
Norge

Håvard Hjulstad
Rådet for teknisk terminologi
Riddervoldsgate 3
N-0258 Oslo 2

Kirsti Rye Ramberg
EDB-tjenesten for humanistiske fag
Universitetet i Trondheim
N-7055 Dragvoll

Victoria Rosén
Institutt for fonetikk og lingvistikk
Universitetet i Bergen
Sydnesplass 9
N-5007 Bergen

Arne S. Svindland
Inst. f. Fonetikk og lingvistikk
Universitetet i Bergen
Sydnesplass 9
N-5027 Bergen

SVERIGE

Lars Ahrenberg
Inst. f. Datavetenskap
Linköpings Universitet
S-581 83 Linköping

Lars Borin
Uppsala Universitet
Centrum för Datorlingvistik
Box 513
S-751 20 Uppsala

Benny Brodda
Stockholms Universitet
Institut f. Lingvistik
S-106 91 Stockholm

Mats Eeg-Oloffson
SWETRA
Institut f. Lingvistik
Lunds Universitet
Helgonabacken 12
S-223 62 Lund

Eva Ejerhed
Institut f. Lingvistik
Umeå Universitet
S-901 87 Umeå

Maria Poporowska Gråstaj
Språkdata
Göteborgs Universitet
S-412 98 Göteborg

Lars Gustafsson
SWETRA
Institut f. allmän
språkvetenskap
Lunds Universitet
Helgonabacken 12
S-223 62 Lund

Anna Sågvall Hein
Språkdata
Göteborgs Universitet
S-412 98 Göteborg

Gunnel Källgren
Stockholms Universitet
Institut f. Lingvistik
S-106 91 Stockholm

Lennart Lönngren
Uppsala Universitet
Centrum f. Datorlingvistik
Box 513
S-751 20 Uppsala

Gudrun Magnúsdottir
Språkdata
Göteborgs Universitet
S-412 98 Göteborg

Valentina Rosén
Centrum f. Datorlingvistik
Uppsala Universitet
Box 513
S-751 20 Uppsala

Klaus Schubert
Uppsala Universitet
FUMS
Box 1834
S-751 48 Uppsala

Bengt Sigurd
Lunds Universitet
Inst. f. Lingvistik och Fonetik
Helgonabacken 12
S-223 62 Lund

Margareta Sjöberg
Centrum f. Datorlingvistik
Uppsala Universitet
S-751 20 Uppsala

Annette Östling Andersson
Romanska Institutionen
Uppsala Universitet
Box 513
S-751 20 Uppsala

DANMARK

Jon Albris
Institut for Nordisk Filologi
Københavns Universitet
Njalsgade 80
DK-2300 København S

Poul Andersen
EUROTRA - DK
Københavns Universitet
Njalsgade 80
DK-2300 København S

Torben Arboe Andersen
Inst. f. Jysk Sprog- og
Kulturforskning
Århus Universitet
Niels Juelsgade 84
DK-8200 Århus N

Annelise Bech
EUROTRA - DK
Københavns Universitet
Njalsgade 80
DK-2300 København S

Frede Boje
EUROTRA - DK
Københavns Universitet
Njalsgade 80
DK-2300 København S

Anna Braasch
EUROTRA - DK
Københavns Universitet
Njalsgade 80
DK-2300 København S

Søren Brandt
Prinsessegade 17 A, 5. th.
DK-1422 København K

Ellen Christoffersen
Handelshøjskole Syd
Tvedvej 9
DK-6000 Kolding

Marianne Dall
EUROTRA - DK
Københavns Universitet
Njalsgade 80
DK-2300 København S

Gert Engel
Handelshøjskole Syd
Grundtvigs Allé 100
DK-6400 Sønderborg

Ruth Feil
Handelshøjskolen i Århus
Tysk Institut
Afdeling for Datalingvistik
Fuglesangs Allé 4
DK-8210 Århus V

Hanne Fersøe
EUROTRA - DK
Københavns Universitet
Njalsgade 80
DK-2300 København S

Steffen Leo Hansen
Institut for Datalingvistik
Handelshøjskolen i København
Howitzvej 60
DK-2000 Frederiksberg

Annette Hartnack
EUROTRA - DK
Københavns Universitet
Njalsgade 80
DK-2300 København S

Hanne Hinz
Institut for Datalingvistik
Handelshøjskolen i København
Howitzvej 60
DK-2000 Frederiksberg

Ebba Hjorth
Gammeldansk Ordbog
Njalsgade 80
DK-2300 København S

Bente Holmberg
Det danske Sprog- og
Litteraturselskabs Ordbøger
Njalsgade 80
DK-2300 København S

Henrik Holmberg
Bysociolingvistik
c/o Inst. f. Dansk Dialekt-
forskning
Njalsgade 80
DK-2300 København S

Henrik Holmboe
Afd. for Datalingvistik
Handelshøjskolen i Århus
Fuglesangs Allé 4
DK-8210 Århus V

Kirsten Høffding
Institut for Datalogi
Roskilde Universitetscenter
Postbox 260
DK-4000 Roskilde

Jane Rosenkilde Jacobsen
Institut for Lingvistik
Københavns Universitet
Njalsgade 80
DK-2300 København S

Steen Jansen
Københavns Universitet
Det humanistiske Edb-center
Njalsgade 80
DK-2300 København S

Niels Jæger
EUROTRA - DK
Københavns Universitet
Njalsgade 80
DK-2300 København S

Merete K. Jørgensen
Gammeldansk Ordbog
Njalsgade 80
DK-2300 København S

Henrik Kersting
Institut for Datalingvistik
Handelshøjskolen i København
Howitzvej 60
DK-2000 Frederiksberg

Sabine Kirchmeier-Andersen
EUROTRA-DK
Københavns Universitet
Njalsgade 80
DK-2300 København S

Poul Søren Kjærsgaard
Institut for Erhvervssprog
Odense Universitet
Campusvej 55
DK-5230 Odense M

Klaus Kjøller
Københavns Universitet
Institut for Nordisk Filologi
Njalsgade 80
DK-2300 København S

Karen M. Lauridsen
Engelsk Institut
Handelshøjskolen i Århus
Fuglesangs Allé 4
DK-8210 Århus V

Bo Laursen
Institut for Fransk
Handelshøjskolen i Århus
Fuglesangs Allé 4
DK-8210 Århus V

Bodil Nistrup Madsen
Institut for Datalingvistik
Handelshøjskolen i København
Howitzvej 60
DK-2000 Frederiksberg

Bente Maegaard
EUROTRA-DK
Københavns Universitet
Njalsgade 80
DK-2300 København S

Erik Møller
Københavns Universitet
Institut for Nordisk Filologi
Njalsgade 80
DK-2300 København S

Anders Nygaard
EUROTRA-DK
Københavns Universitet
Njalsgade 80
DK-2300 København S

Eva Nørreslet
EUROTRA-DK
Københavns Universitet
Njalsgade 80
DK-2300 København S

Stig Örjan Ohlsson
Københavns Universitet
Det humanistiske Edb-center
Njalsgade 80
DK-2300 København S

Jørgen Olsen
Københavns Universitet
Det humanistiske Edb-center
Njalsgade 80
DK-2300 København S

Carsten Kruse Olsson
EUROTRA-DK
Københavns Universitet
Njalsgade 80
DK-2300 København S

Susanne Nøhr Pedersen
EUROTRA-DK
Københavns Universitet
Njalsgade 80
DK-2300 København S

Torsten Pedersen
Institut for Datalogi
Roskilde Universitetscenter
Postbox 260
DK-4000 Roskilde

Margrethe Petersen
Institut for Engelsk
Handelshøjskolen i Århus
Fuglesangs Allé 4
DK-8210 Århus V

Pia Riber Petersen
Dansk Sprognævn
Njalsgade 80
DK-2300 København S

H. Picht
Institut for Spansk
Handelshøjskolen i København
Howitzvej 60
DK-2000 Frederiksberg

J. Qvistgaard
Institut for Fransk
Handelshøjskolen i København
Fabrikvej 7
DK-2000 Frederiksberg

Ebbe Spang-Hanssen
Romansk Institut
Københavns Universitet
Njalsgade 80
DK-2300 København S

Henning Søndergaard
Tysk Institut
Handelhøjskolen i Århus
Fuglesangs Allé 4
DK-8210 Århus V

Ole Togeby
EUROTRA - DK
Københavns Universitet
Njalsgade 80
DK-2300 København S

Helle Wegener
Institut for Datalingvistik
Handelshøjskolen i København
Howitzvej 60
DK-2000 Frederiksberg

Henrik Prebensen
Københavns Universitet
Det humanistiske Edb-center
Njalsgade 80
DK-2300 København S

Hanne Ruus
Institut for Nordisk Filologi
Københavns Universitet
Njalsgade 80
DK-2300 København S

Uffe Sonne Svendsen
EUROTRA - DK
Københavns Universitet
Njalsgade 80
DK-2300 København S

Torben Thrane
Københavns Universitet
Det humanistiske Edb-center
Njalsgade 80
DK-2300 København S

Carl Vikner
Institut for Datalingvistik
Handelshøjskolen i København
Howitzvej 60
DK-2000 Frederiksberg

Henning Ørum
Københavns Universitet
Det humanistiske Edb-center
Njalsgade 80
DK-2300 København S

DELTAGERLISTE

DATAMATSTØTTET LEKSIKOGRAFI OG TERMINOLOGI

5.-6.11 1987

FINLAND

Centralen för Teknisk
Terminologi
Elisabetsgatan 16 D
SF-00170 Helsingfors

Erja Nikunen
Forskningscentralen för
de inhemska språken
Elisabetsgatan 16 C 21
SF-00170 Helsingfors

Krista Varantola
Department of English
University of Turku
SF-20500 Turku

Simo Merne
School of Translation Studies
Turku University
Aurakatu 11
SF-20100 Turku

Tuomo Tuomi
Forskningscentralen för
de inhemska språken
Elisabetsgatan 16 A 1
SF-00170 Helsingfors

Gunvor Wikberg-Penttilä
Forskningscentralen för
de inhemska språken
Elisabetsgatan 16 C 21
SF-00170 Helsingfors

NORGE

Håvard Hjulstad
Rådet for Teknisk Terminologi
Riddervoldsgate 3
N-0258 Oslo 2

Victoria Rosén
Institutt for Fonetikk
og Lingvistikk
Universitetet i Bergen
Sydnesplass 9
N-5007 Bergen

Kirsti Rye Ramberg
EDB-tjenesten for
humanistiske fag
Universitetet i Trondheim
N-7055 Dragvoll

Arne S. Svindland
Universitetet i Bergen
Inst. for Fonetikk og
Lingvistikk
Sydnesplass 9
N-5027 Bergen

Ivar Utne
Norsk Termbank
Strömgate 53
N-5007 Bergen

SVERIGE

Eva Ejerhed
Inst. för Lingvistik
Umeå Universitet
S-901 87 Umeå

Maria Poporowska Grånåstaj
Språkdata
Göteborgs Universitet
S-412 98 Göteborg

Lennart Lönngren
Centrum för Datorlingvistik
Box 513
Uppsala Universitet
S-751 20 Uppsala

David Mighetto
Institutionen för Romanska Språk
Spanska Avd.
Universitetet
S-412 98 Göteborg

Klaus Rossenbeck
Lunds Universitet
Tyska Institutionen
Helgonabacken 14
S-223 62 Lund

Gudrun Magnúsdottir
Språkdata
Göteborgs Universitet
S-412 98 Göteborg

Valentina Rosén
Centrum för Datorlingvistik
Uppsala Universitet
Box 513
S-751 20 Uppsala

Margareta Sjöberg
Centrum för Datorlingvistik
Uppsala Universitet
Box 513
S-751 20 Uppsala

DANMARK

Jon Albris
Københavns Universitet
Institut for Nordisk Filologi
Njalsgade 80
DK-2300 København S

Frede Boje
EUROTRA - DK
Københavns Universitet
Njalsgade 80
DK-2300 København S

Søren Brandt
Prinsessegade 17 A, 5. th.
DK-1422 København K

Grete Duvå
Handelshøjskolen i Århus
Fransk Institut
Fuglesangs Allé 4
DK-8210 Århus V

Gert Engel
Handelshøjskole Syd
Grundtvigs Allé 100
DK-6400 Sønderborg

Annelise Grinsted
Handelshøjskole Syd
Østervang 2
DK-6800 Varde

Torben Arboe Andersen
Århus Universitet
Institut for Jysk Sprog- og
Kulturforskning
Niels Juelsgade 84
DK-8200 Århus N

Anna Braasch
EUROTRA - DK
Københavns Universitet
Njalsgade 80
DK-2300 København S

Anne Duekilde
ODS-Supplementet
Njalsgade 80
DK-2300 København S

Gunhild Dyrberg
Institut for Fransk
Handelshøjskolen i København
Fabrikvej 7
DK-2000 Frederiksberg

Ruth Feil
Handelshøjskolen i Århus
Tysk Institut
Afd. for Datalingvistik
Fuglesangs Allé 4
DK-8210 Århus V

Inge Gorm Hansen
Institut for Engelsk
Handelshøjskolen i København
Fabrikvej 7
DK-2000 Frederiksberg

Steffen Leo Hansen
Institut for Datalingvistik
Handelshøjskolen i København
Howitzvej 60
DK-2000 Frederiksberg

Bent Hauschildt
Handelshøjskolen i Århus
Tysk Institut
Fuglesangs Allé 4
DK-8210 Århus V

Hanne Hinz
Institut for Datalingvistik
Handelshøjskolen i København
Howitzvej 60
DK-2000 Frederiksberg

Ebba Hjorth
Københavns Universitet
Gammeldansk Ordbog
Njalsgade 80
DK-2300 København S

Bente Holmberg
Det danske Sprog- og
Litteraturselskabs Ordbøger
Njalsgade 80
DK-2300 København S

Henrik Holmberg
Bysociolingvistik
c/o Institut for Dansk
Dialektforskning
Njalsgade 80
DK-2300 København S

Henrik Holmboe
Handelshøjskolen i Århus
Afdeling for Datalingvistik
Fuglesangs Allé 4
DK-8210 Århus V

Jane Rosenkilde Jacobsen
Københavns Universitet
Institut for Lingvistik
Njalsgade 96
DK-2300 København S

Merete K. Jørgensen
Københavns Universitet
Gammeldansk Ordbog
Njalsgade 80
DK-2300 København S

Poul Søren Kjærsgaard
Odense Universitet
Institut for Erhvervssprog
Campusvej 55
DK-5230 Odense M

Karen M. Lauridsen
Handelshøjskolen i Århus
Engelsk Institut
Fuglesangs Allé 4
DK-8210 Århus V

Bo Laursen
Handelshøjskolen i Århus
Institut for Fransk
Fuglesangs Allé 4
DK-8210 Århus V

Birgitte Lauterbach
Institut for Fransk
Handelshøjskolen i København
Fabrikvej 7
DK-2000 Frederiksberg

Bodil Nistrup Madsen
Institut for Datalingvistik
Handelshøjskolen i København
Howitzvej 60
DK-2000 Frederiksberg

Else Marker-Larsen
Institut for Fransk
Handelshøjskolen i København
Fabrikvej 7
DK-2000 Frederiksberg

Bodil Martinsen
Handelshøjskolen i Århus
Fransk Institut
Fuglesangs Allé 4
DK-8210 Århus V

Erik Møller
Københavns Universitet
Institut for Nordisk Filologi
Njalsgade 80
DK-2300 København S

Ole Norling-Christensen
Utterslevvej 7, 1. th.
DK-2400 København NV

Torsten Pedersen
Institut for Geografi, Samfunds-
analyse og Datalogi
Roskilde Universitetscenter
Hus 20.2
P.B. 260
DK-4000 Roskilde

Margrethe Petersen
Handelshøjskolen i Århus
Engelsk Institut
Fuglesangs Allé 4
DK-8210 Århus V

Pia Riber Petersen
Dansk Sprognævn
Njalsgade 80
DK-2300 København S

H. Picht
Institut for Spansk
Handelshøjskolen i København
Howitzvej 60
DK-2000 Frederiksberg

Svend O. Poulsen
Handelshøjskolen i Århus
Tysk Institut
Fuglesangs Allé 4
DK-8210 Århus V

Hanne Puggaard
Institut for Spansk
Handelshøjskolen i København
Howitzvej 60
DK-2000 Frederiksberg

Birte Qvistgaard
Sektionschef i EF-Kommissionen
Terminologi og EDB-anvendelser
CCE
200, rue de la Loi
B-1049 Bruxelles

J. Qvistgaard
Institut for Fransk
Handelshøjskolen i København
Fabrikvej 7
DK-2000 Frederiksberg

Anne-Mette Keogh Rasmussen
Terminologiafdelingen
Handelshøjskolen i København
Howitzvej 60
DK-2000 Frederiksberg

Hans-Otto Rosenbohm
Odense Universitet
Institut for Erhvervssprog
Campusvej 55
DK-5230 Odense M

Hanne Ruus
Københavns Universitet
Institut for Nordisk Filologi
Njalsgade 80
DK-2300 København S

Regina Saroléa
Chef for glossargruppen i EF
Terminologi og EDB-anvendelser
CCE
200, rue de la Loi
B-1049 Bruxelles

Henning Søndergaard
Handelshøjskolen i Århus
Tysk Institut
Fuglesangs Allé 4
DK-8210 Århus V

Joan H. Tournay
Institut for Fransk
Handelshøjskolen i København
Fabrikvej 7
DK-2000 Frederiksberg

Tor Valén
Århus Universitet
Niels Juelsgade 84
DK-8200 Århus N

Carl Vikner
Institut for Datalingvistik
Handelshøjskolen i København
Howitzvej 60
DK-2000 Frederiksberg

Helle Wegener
Institut for Datalingvistik
Handelshøjskolen i København
Howitzvej 60
DK-2000 Frederiksberg

Lotte Weilgaard
Handelshøjskole Syd
Østervang 2
DK-6800 Varde