GELEXI PROJECT: PROLOG IMPLEMENTATION OF A TOTALLY LEXICALIST GRAMMAR

GÁBOR ALBERTI / KATA BALOGH / JUDIT KLEIBER

Department of Linguistics
University of Pécs
H-7624 Pécs, Ifjúság útja 6.
albi@btk.pte.hu /matrica@stud.btk.pte.hu / kleiberj@btk.pte.hu
aalberti@axelero.hu / balzso@axelero.hu / kleiberj@index.hu

A new sort of generative grammar — *GASG* (Generative/Generalized Argument Structure Grammar) — is demonstrated in this paper (together with its Prolog implementation); which is more consistently and radically lexicalist than any earlier generative grammar: the lexicon plays the role of the generating component (→ *Ge(nerative) Lexi(con) Project).*[1]

The first motivation of the enterprise (Alberti 1990, 1996) was the stubborn problem of compositionality in Discourse Representation Theory (e.g. Kamp 1981). Nowadays (van Eijck – Kamp 1997; see also Karttunen 1986, Zeevat 1991a-b) some kind of unificational Categorial Grammar is held to promise the best chance for capturing the language→DRS transition in a properly compositional manner (according to the authors mentioned). The basic problem with UCG, which has amounted to the starting-point of GASG, lies in the fact that syntax, deprived of the information concerning sentence cohesion in favor of the unification mechanism and reduced to the primitive task of combining adjacent words, will produce linguistically irrelevant constituents (Karttunen 1986: 19; see his examples). Instead of the usual categorial apparatus, we argue that adjacency and order among words are to be treated by the same technique of unification as morphological cohesion (agreement, case). And the engine combining words to form sentences must be unification itself, which is capable of running Prolog programs properly.

The crucial point of our proposal is the introduction of rank parameters in the lexical formulas (substituting for the usual categorial apparatus) where adjacency requirements between words are fixed and the special simultaneously recursive definition of (the satisfaction of) these formulas.

1. Arguments for the Principle of Total Lexicalism and GASG

1.1 A Metatheoretical Argument. We are arguing (see also Alberti 1999) that studying the possibility of a totally lexicalist grammar is not only a legitimate research program but an unavoidable metatheoretical task forced upon us by the five-decade scientific tendency in the course of which the Lexicon is occupying more and more areas at the loss of syntax in *every* important branch of the family of generative theories.

We are referring to the tendency in the course of which *generative theory*, which appeared in the fifties as a radically syntax-centered linguistic theory with a very simple lexicon (Chomsky 1957), had become by the nineties — though separated into several branches (MP, CGs, LFG, HPSG, TAG, C&S)— a theory with a highly reduced syntax and a lexicon of rich content and structure. Whenever a new non-Chomskyan branch was founded, leading points of the program were almost always the extension and a more exact and thorough formalization of the area of the lexicon, and a definite ambition to store the information concerning the syntax–semantics interface in the lexicon. In the light of these facts, the radical lexicon-centrism and the non-language-specific approach to syntax characteristic of the Chomskyan Minimalist Program (Chomsky 1995) are of even greater importance: “The syntactic engine itself – the autonomous principles of composition and manipulation Chomsky now labels ‘the computational system’ – has begun to fade into the background. Syntax reduces to a simple description of how constituents drawn from the lexicon can be combined and how movement is possible (i.e. how something other than the simple combination of independent constituents is possible). The computational system, this simple system of composition, is constrained by a small set of economy principles which Chomsky claims enforce the general requirement, ‘do the most economical things to create structures that pass the interface conditions (converge at the interfaces)” (Marantz 1995: 380, section 8 *The End of Syntax).*

Our conclusion drawn from this metatheoretical discussion is that it is a grammar dispensing with Merge and Move that is derivable from the general generative philosophy as a conceptual minimum (Chomsky 1995). GASG is nothing else than an attempt to realize this conceptual minimum: to get rid of Move, to reduce Merge to a non-PS-tree-producing unification, and to store all linguistic information in the lexicon (the minimalist feature checking is also a kind of unification, so unification seems to be the final “inevitable coceptual minimum”).[2]

---

[1]Special thanks are due to the Hungarian National Scientific Research Fund (grant no. OTKA F026658), the Széchenyi Professorial Scholarship (G. Alberti 1999–2002), the Faculty of the Humanities of the University of Pécs and the Rector of the university for their financial support. Futher, we are grateful to Henk Zeevat and other participants of the 4th Tbilisi Symposium on Language, Logic and Computation (held in Borjomi) for their valuable comments on our talk this article is based on.

[2] It is worth quoting here Dowty’s (1996: 12, 53) opinion from *Toward a minimalist theory of syntax* about the legitimacy of PS trees, which Chomsky (1995: 403) seems to consider “inescapable on the weakest interface conditions”: “I suspect syntacticians today have almost come to think of the “primary empirical data” of syntactic research as phrase structure trees, so firm are our convictions as to what the right S-structure tree for most any given sentence is. But speakers of natural languages do not speak trees, nor do they write trees on paper when they communicate. The primary data for syntax are of course only *strings* of words, and everything in syntactic description beyond that is part of a theory, invented by a linguist.” The author’s aim is “getting linguists to question our automatic assumptions about constituents and our basis for assuming as a methodological principle that languages must always have a phenogrammatical syntactic structure describable by phrase structure trees.”

1.2 A Theoretical Argument. The main theoretical argument in favor of GASG is that it promises a better answer to the stubborn problem of *compositionality* as to the morphosyntax→DRS (Discourse Representation Structure; e.g. Kamp 1981) transition than PSGs (Phrase Structure Grammars). We argue (Alberti 1999) that the failure of elaborating a properly compositional solution to this language→DRS transition arises from the fundamental incompatibility of the strictly hierarchically organized generative syntactic phrase structures with the basically unordered DRSs (or ones ordered but in an entirely different way).

Nowadays (Zeevat 1987, Karttunen 1986, van Eijck and Kamp 1997) some kind of Categorial Grammar (CG) is held to promise the best chance for capturing the language→DRS transition in a properly compositional manner: a version of (Classical) Categorial Grammar with capacity increased[3] by the technique of *unification,* applied in Prolog, for instance *(UCG)*[4]. The basic problem with UCG, which has amounted to the starting-point of GASG, lies in the fact that syntax, deprived of the information concerning sentence cohesion in favor of the unification mechanism and reduced to the primitive task of combining adjacent words, will produce linguistically irrelevant constituents. According to Karttunen's (1986: 19) remark on UCG trees: they look like PS trees but they are only "analysis trees"; and he adds "all that matters is the resulting [morphological] feature set."

This general problem with Unificational CGs requires more elaboration here; which will be based on Karttunen's (1986) paper on Finnish word order variation and long-distance dependencies.[5]

In Karttunen's grammar the linguistically relevant "checking" among words relies on a mechanism in the course of which attribute–value paths attached to categories are to be unified: "The value of *argument* is unified with the feature set of the expression the functor attempts to combine with. If this unification fails, the two expressions cannot be combined; if it succeeds, the feature set of the consequent expression is the same as the value of the functor's *result*." Categories (e.g. NPs) and the categorial rules of their combination are assumed to be practically trivial: "We define a nominative [/accusative] NP in Finnish as an expression that combines with an adjacent verb phrase to yield a verb phrase (V|V) in which it plays the role of the subject [/object]." "They [these expressions of category V|V] combine with a basic verb phrase to yield a verb phrase that is the same as the original except that the content of the functor has been merged with the argument in a manner determined by the functor. This latter aspect of the analysis of course can only be expressed in a unification–based formalism. … The parsing algorithm can be extremely simple: *Always try to combine a functor with an adjacent argument.*"

The advantage of UCG lies in the fact that "…by treating subject and object noun phrases syntactically as functors we have accounted for [some generalizations about the free Finnish word order] by using only Function Application. In a CG of a more traditional sort, in which the verb is responsible for linking up with its syntactic dependents, type raising and function composition must be introduced to achieve the same result."

There is a serious (linguistic and computational) disadvantage, however, discussed by Karttunen himself: "It is convenient to represent the analyis of a phrase as a tree that shows how the resulting feature set was derived. However, the structure of the analysis tree has no linguistic significance in our system: in this respect analysis trees are different from PS trees as they are traditionally construed in linguistics. *All that matters is the resulting feature set.* Because no functor has any priority over others with respect to order of application, the same result can often be obtained in more than one way. This is potentially troublesome from a computational point of view. … From the parser's point of view, this is a "spurious ambiguity" because the alternative analyses yield exactly the same set of features. In a more complicated sentence, spurious ambiguities multiply very quickly…"

As for the way out, the author's final conclusion / conjecture is that "…a *radical lexicalist* approach is a better alternative." GASG is to be regarded as an attempt to reach the most radically —*totally*— lexicalist solution, which can be formulated informally as follows: GASG = UCG – C where the "subtrahend" refers to the categorial way(s) of combining words (i.e. Function Application in UCG), and 'C' refers to the (constituent combining) apparatus of categorial grammars.

---

[3] The problem with Classical CG is that it has only a context free generative capacity (according to the Chomsky hierarchy of grammars), which is proved (Shieber 1985) to be insufficient for the description of human languages. There seem to be two ways to increase the generative capacity of CCG: to let in, in opposition to the original goals, a few combinatorial means (though non-language-specific ones such as Function Composition ($X/Y\ \ Y/Z \Rightarrow X/Z$), Commutativity ($(X/Y)\backslash Z \Rightarrow (X\backslash Z)/Y$), Type Raising ($X \Rightarrow Y/(Y\backslash X)$)) in addition to the "classical" Function Application ($X/Y\ \ Y \Rightarrow X$), or to introduce the technique of *unification*, applied e.g. in Prolog (UCG).

[4] The anonymus reviewer of this paper mentioned Muskens' "lambda" DRT as a popular option at this point; (s)he aslo mentioned further relevant alternatives to our approach (Assumption Grammars, Link Grammar, Dependency Grammars etc.). The size of this paper does not make it possible for us to provide a comparison of our system with these (and further) alternatives – we hope, however, that we will get the chance for doing that in the future. Special thanks are due to the reviewer for these useful comments helping in deciding the direction of our future research.

[5] The relevant property of Finnish (in which respect this language is similar to Japanese or Hungarian) is its ("highly") free word order, concomitant with a rich morphology.

2. GASG

GASG can be characterized generally as a distinguished monostratal grammar where there is no need for a "separate generating engine" (linguistic rules of computation) at all. The engine must be *unification* itself, which is capable of running Prolog programs properly. The rich description of a lexical sign (say, out of a group of lexical signs selected from the Lexicon in order to combine them to form a sentence) serves a double purpose. It characterizes the *potential environment* of the given sign in possible grammatical sentences in order for the sign to find the morphologically (or in other ways) compatible elements and to avoid the incompatible ones in the course of forming a sentence. And the lexical description characterizes the sign itself in order for other words to find (or not to find) it, on the basis of similar "environmental descriptions" belonging to the lexical characterizations of these other words. And while the selected words are seeking each other on the basis of their formal features suitable for unification, their semantic features are also being unified simultaneously; so by the end of a successful building it will have been verified that a particular sequence of fully inflected words constitutes a grammatical sentence, and its semantic representation, a DRS, will also have been built at our disposal.

As the special treatment of word order and the Prolog implementation of GASG are intended to be concentrated on in this paper, there is no place here to provide a detailed general demonstration. We can only sketch such an important topic as the morphological basement of GASG. Here we offer the two papers serving as the basis of this short summary: Alberti (2000b, 2001). The starting-point is a comparative analysis of the German, Hungarian and English version of a sentence:

(1) a. Mein kluger Lehrer fand eine freundliche ungarische Studentin.
my-m-sg-NOM clever-m-sg-NOM teacher found-3sg an-f-ACC friendly-f-sg-ACC Hung.-f-sg-ACC she-student
b. Az én bölcs tanár-om talál-t egy barátságos magyar diáklány-t.
the I clever teacher-poss-1sg found-3sg a friendly Hungarian student-girl-acc
c. My clever teacher found a friendly Hungarian student.

The annotations belonging to the German and the Hungarian (but not the English) words refer to a rich system of morphological "sensitivity" among words in sentences (practically case and agreement relations; see Lehmann (1988)). The task of this system can be formulated by means of a comparison between the (simplified common) DRS of the sentences, shown in (2a) below, and the "proto-DRS" in (2b), which is the result of collecting the (discourse–) semantic contributions of the lexical items evoked by the words of the German sentence (still regarded as independent elements).

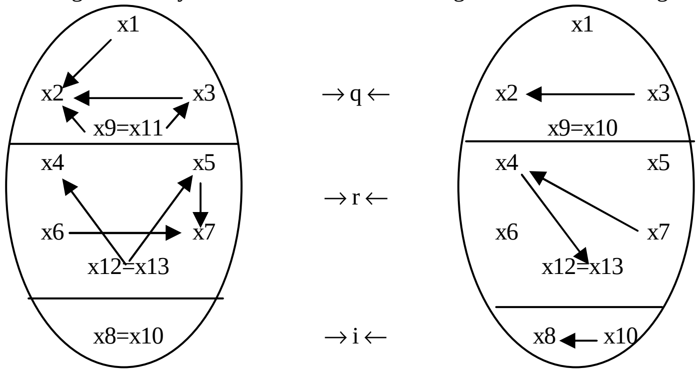
(2) a. q ∧ i ∧ r ∧ e ∧ clever(q) ∧ teacher-of(q,i) ∧ I(i) ∧ find(e,q,r) ∧ past(e) ∧ friendly(r) ∧ Hung.(r) ∧ student(r)
b. klug(x1), Lehrer(x2,…), fand(x3,x4), freundlich(x5), ung.(x6), Stud.(x7)
mein: x8=i, x9, X(x9,x10), x8=x10, Y(x11,…), x9=x11
eine: x12, W(x12), Z(…,x13,…), x12=x13

In a proto-DRS, thus, (proto–) referents coming from different lexical items are to be different. Now the task of the infrasentential morphological system can be formulated as follows: it should provide information enough for making the "underspecified" proto-DRS specific, by converting it into a "normal" DRS. We argue (Alberti 2000b, 2001) that this procedure is based on the following universal: if two words belonging to two different lexical items in a sentence stand in some morphological / morphosyntactic relation (agreement or case relation), then the corresponding lexical items stand in a *copredicative* relation (i.e. there are referents to be identified in the (underspecified) propositions constituting the semantic components of the given lexical items).

Arrows in the left figure in (3) below show the proto-referents in (2b) that can be identified due to some morphological sensitivity between words. The agreement between *kluger* and *Lehrer* in number, gender and case, for instance, indicates that the teacher coincides with the person claimed to be clever, which can be expressed by the identity *x1=x2.*[6] This relation on proto-referents can be regarded as a potential *generator system* of the equivalence relation represented by the final (specified) DRS — in the sense that the latter relation is the reflexive–symmetric–transitive closure (Partee *et al.* 1990) of the former one in the ideal case.

[6] Doubtful instances of copredication are discussed (interpreted appropriately) in Alberti (2000a).

(3) Concordial generator systems: sufficient German cgs. / insufficient Hungarian cgs.



Only the German sentence version exhibits this ideal case in (1), however; which can be formulated by saying that the concordial generator system that belongs to sentence (1a) is *sufficient.* The cgs. belonging to the Hungarian sentence in (1b) is *insufficient* in the same sense (because of the lack of any kind of agreement between attributive adjectives and nouns in Hungarian), and the corresponding English cgs. is simply empty. These latter two facts suggest that we should "re-discover" syntax:

A copredicative relation between two lexical items in a sentence can be expressed in human languages also by *pure syntactic relations* (e.g. adjacency and/or (immediate) precedence; see fn. 2 again) between (the) two words belonging to them. Pure syntactic relations and morphological dependence relations thus are both *formal intrasentential relations* between words suited for marking copredicative (semantic) relations between the corresponding lexical items.

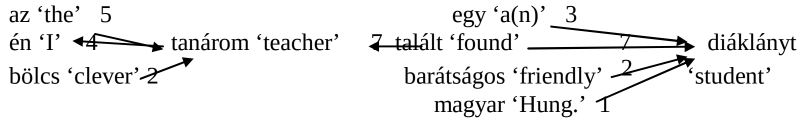
In lexical descriptions of GASG these two kinds of formal relations are represented in the same way, on the same level; there is no separation such as that in (U)CGs (and other PSGs), where unifiable attribute–value paths (or feature systems) are responsible for morphological relations and some categorial apparatus of combinatorial rules (or Merge and Move rules) are responsible for word order. The lexical item of *talált* 'found' in (1c) serves as an illustration (see Alberti 2000b, 2001):

(4) Λ5 =< {v5 = *talált},*
{fin.past(v5), verb.tr(v5), noun(α, V5.11), 3.sg(α, V5.11), arg(α, nom, V5.11, v5), ref(α, V5.12),
immprec(**α=5**, V5.12, V5.11),
immprec(**α=7,** V5.11, v5), noun(α, V5.21), ref.indef(α, V5.22), 3(α, V5.21), arg(α, acc, V5.21, v5),
immprec(**α=5**, V5.22, V5.21), immprec(**α=7,** v5, V5.21)},
{∧(α, found(X5.1, X5.2), Q5.1(X5.1)), ∧(α, found(X5.1, X5.2), Q5.2(X5.2))},
{corr(v5, found), corr(V5.11, Q5.1), corr(V5.21, Q5.2)}>

Only the most relevant details of lexical items (and this particular one) will be explained here. Lexical items consist of four components, out of which the first one shows the *own word:* a fully inflected word to look for in sentences whose parsing is supposed to require the given lexical item. The second component provides the morphosyntactic characterization of the *own word* (denoted by a lower case letter in harmony with the practice in Prolog where constants are marked in this way) and the *environmental words* — they are variables to be unified with own words of other lexical items. In the example above own word v5 is characterized as a transitive verb in the past tense and there are four environmental words because of the two arguments of 'find.' Why are there four environmental words? As categories, such as DP, are not intended to be referred to, a nominal argument should typically be referred to via its two "pillars": the nominal pillar (e.g. 'teacher': V5.11) and the "referential" pillar (the one responsible for the (usually obligatory) referentiality of the argument (Alberti 1997); e.g. 'the': V5.12). If *Peter found sy,* the two pillars of the subject coincide, i.e. both are to be unified with one and the same own word, the one belonging to the lexical item of 'Peter.' As for the two pillars of the potential object of 'found,' a crucial morphosyntactic property of the nominal pillar (V5.21) is its accusative case, and the referential pillar (V5.22) is characterized as an indefinite element (with respect to the definite conjugation of the finite verb).

The contribution of a lexical item to the determination of sentential word order basically lies in the *immprec* ('immediate precedence') requirements, whose satisfaction is defined so that the α *rank parameters* (see above in (4)) are considered too. Section 3 is devoted to this topic, so only a problem is mentioned here: if all *immprec* requirements coming from the lexical items that the parsing of the given sentence is based on are considered, the following contradictory requirements will have been collected (Alberti 2000b, 2001), where the formula *'the'---5--→'teacher,'* e.g., means that in one of the relevant lexical items there is a condition of rank 5 requiring the referential pillar of a "DP" to immediately precede the nominal pillar:

(5)



The main point in Section 3 will be the claim that requirements like the particular one above is *not violated* in sentence (1b) in spite of the word order '*the* I clever *teacher*…' but it is satisfied *indirectly* by satisfying stronger requirements (those of higher ranks denoted by smaller numbers).

To finish demonstrating the lexical item, the third component provides its discourse–semantic contribution to the DRS of the sentence parsed, and the fourth component is responsible for fixing the correspondence between elements of the morphosyntactic component (own and environmental words) and those of the discourse–semantic component (own and environmental predicates).

Computations in the course of which variables (of "word," "predicate" and other types) of lexical items belonging to particular (Hungarian, English, German and Dutch) sentences to be parsed are unified with constants (own words and own predicates) of each other (resulting in some evaluation of grammaticality and specified DRSs) are demonstrated in the following papers: Alberti (1998, 1999, 2000b, 2001).[7]

3. Rank Parameters in GASG

This short but very important section is devoted to the definition of (the satisfaction of) *immprec* relations (see (4-5) above, which the facility for dispensing with the concept of *constituents* depends on.

(6) DEF.: An *immprec$_n$* relation (immediate precedence relation of rank n) between words w1 and w2 in a sequence of words is satisfied if

1) w1 immediately precedes w2 (indeed), or
2) there is a word w3 (between w1 and w2)[8] such that w3 is to satisfy an immprec$_k$ relation with w1 or w2 where n≥k, and sequences [w1,...,w3] and [w3,...,w2] are *legitimate.*

A sequence [x1,...x2] of words is *legitimate* if

1) it consists of only two members, or
2) there is a word x3 in [x1,...x2] such that, for some number m, x3 is to satisfy an immprec$_m$ relation with x1 or x2, and sequences [x1,...,x3] and [x3,...,x2] are *legitimate.*

The informal content of the definition is that a requirement of rank n concerning the immediate precedence of word w1 relative to w2 can be satisfied either *directly* — by the fact that w1 does immediately precede w2 indeed — or indirectly — by permitting certain words to be inserted between w1 and w2, those, and only those, whose immprec requirement to w1 or w2 is stronger, or are dependents of such words, or dependents of dependents, or dependents of dependents of dependents, etc[9]. Sequences of such "dependent" words are called *legitimate,* and this relation should be checked either trivially — by recognizing that the list of words in question only consists of the two peripheral members, and hence nothing is inserted inside, or by pointing out novel and novel "dependents of dependents."

---

[7] Certain lexical items are supposed to have more than one own word (preverb+verb constructions (Alberti 1999), idioms (Alberti 2001)) or no own word (Hungarian focus (Alberti 1998), English interrogative operator (Alberti 2000b)).

[8] The version with parentheses is to be applied in the case of Swiss German cross-serial dependencies (see Alberti (2000b) where this theoretically very important (Shiebr 1985) construction is compared to German nested dependencies and the corresponding (regular) English nifinitival constructions).

[9] This point seems to be the best one for answering a question of the anonymus reviewer's on the treatment of long distance dependencies in GASG. The article on this topic (Alberti 2002) is unfortunately in Hungarian. The essence, however, is clearly demonstrated by (the "recursive" spirit of) the definition in (6) above. What is to be captured is the *unbounded* distance between the "real" lexical regent of an argument and its "surface" place in the neighborhood of a higher regent, which is 'higher' in the sense that the former regent is an argument of an argument of an argument… of the latter regent. One mugth think that it is hard to formalize this unbounded functional relation in a theory based on the lexical description of "local" environmental requirements of words. The recursive technique of definition, however, clearly proves the opposite: we may define the "higher regent" of an element as either its immediate (lexical) regent or the higher regent of its immediate regent. Even constrains on these chains of regents (e.g. some kind of *that* trace filter) are easy to formulate by demanding certain kinds of regent-argument relations.

The $immprec_5$ requirement between 'the' and 'teacher' in (5), for instance, is satisfied indirectly in sentence (1b): (possessive) 'I' is inserted between them legitimately due to its adjacency requirement of rank 4, then the ['the,' 'I'] sequence is trivially legitimate, and the ['I,' 'clever,' 'teacher'] sequence is legitimate due to the $immprec_2$ requirement between 'clever' and 'teacher'; and finally two trivially legitimate (two-member) sequences should be pointed out. More complicated cases will be analyzed in the course of discussing the Prolog implementation of GASG.

Examples in (7) below illustrate that in Hungarian, but not in English, certain free adverbs ('yesterday,' 'in the library') can be inserted between the finite verb and its arguments quite freely (in the case of idioms as well). This can be accounted for in GASG easily — by choosing the same rank parameter, namely 7, for both the regent–argument adjacency requirement and that between free adverbs and the finite element of sentences, at least in Hungarian. In English, however, the regent–argument adjacency requirement is to be qualified as stronger.

(7) a. Adt-am tegnap Mari-nak a könyvtár-ban egy cikk-et.
gave-1sg yesterday Mary-dat the library-ine a paper-acc
'Yesterday I gave Mary a book in the library.'
b. *I gave yesterday Mary in the library a paper.
c. *Peter kicked yesterday the bucket.
d. Péter be-adta tegnap a kulcs-ot.
Peter in-gave yesterday the key-acc
'Yesterday Peter kicked the bucket.' (lit. 'P. returned the key.')

This also shows that GASG is suitable for the treatment of "rigid" word order as well. Differences between languages can be captured by fixing different adjacency (and order) numeral rank parameters, which is the simplest and most elegant way of expressing parametric differences between human languages (as alternative possible realizations of UG).

The final comment in this section concerns the possibility for applying rank parameters in semantics as well. The ambiguity of the sentence in (8a) below, for instance, can be accounted for by fixing the same rank parameter, in semantics, for adjacency requirements between certain pieces of DRS formulas in the lexical description of *every* and *a(n);* and in this way two final DRS formulas can be computed (see (8a-b) below). Details are available in Alberti (1999).

(8) a. *Every* English boy visited *a* pretty Dutch girl.
b. ...∧**((x4 ∧ boy(x4)) ∧ english(x4)) →** [*(((x5 ∧ girl(x5)) ∧ dutch(x5)) ∧pretty(x5))* ∧ visit(x3, x4, x5)]
c. ...∧*(((x5 ∧ girl(x5)) ∧ dutch(x5)) ∧pretty(x5))* ∧ [**((x4 ∧ boy(x4)) ∧ english(x4))→** visit(x3, x4, x5)]

4. Implementation in Prolog

4.1 Purposes and parsing. As many others, we regard it as a primary purpose of an implementation work to verify the descriptive adequacy of a given theory as well as to make the theory and its practical realization to be suitable for revealing more about languages and their function.

Our program is permanently being developed, and the version that is available now can parse uncompound neutral Hungarian sentences.[10] It is practically to be regarded as a "toy grammar" which can interpret just a small — but non-trivial — fragment of the Hungarian language. In our parser we insist on the theoretically "clear" principles, but naturally we have to make some technical change because of the special features of programming in Prolog. Hence, parts of lexical items in GASG are stored in different places in the programme. The database section contains the own word and inherent properties that lexical items can reveal about themselves. Environmental conditions and properties of words that a lexical item searches are put down in **`relation`** predicates. This part means the "syntax" together with a checking section that contains the **`immprec`** relations (6). The third part of a lexical item — which is semantics — is represented in **`sem`** predicates, and the fourth part of "theoretical" lexical items requires no separate expression in the programme, however, because in our Prolog clauses semantic relations are already written as consequences of morphosyntactic ones.

[10] In this section the parts of the program are written by courier fonts

In our program all lexical items have two main components: the first one is the “own word” and the second one is a label with “own properties.” This label itself also consists of three components, out of which the first one is the English “translation” of the root of the verb. In the case of the second one there are various possibilities: if the given lexical item is a noun, this component contains its case, if a finite verb, the own features, and if a determiner, its definiteness feature. The third component of labels consits of the features that the word “shows” about its arguments: inflection, agreement relations etc.

The clauses from database section below are examples for lexical items:

(9) a. Adjuncts:
```
lexitem("holland",adj("Dutch")).
lexitem("barátságos",adj("friendly")).
```
b. Determiners
```
lexitem("egy",det("a","indef")).
lexitem("a",det("the","def")).
lexitem("három",det("three","indef")).
```
c. Nouns:
```
lexitem("Péter",ln("Peter","NOM",arg(f("sg","3"),s("none","0","none")))).
lexitem("Péter",ln("Peter","POSS",arg(f("sg","3"),s("none","0","none")))).
lexitem("Péternek",ln("Peter","DAT",arg(f("sg","3"),s("none","0","none")))).
lexitem("Péternek",ln("Peter","GEN",arg(f("sg","3"),s("none","0","none")))).
lexitem("lányt",ln("girl","ACC",arg(f("sg","3"),s("none","0","none")))).
lexitem("fiúra",ln("boy","OBL",arg(f("sg","3"),s("none","0","none")))).
lexitem("te",ln("you","NOM",arg(f("sg","2"),s("none","0","none")))).
lexitem("barátom",ln("friend","NOM",arg(f("sg","3"),s("sg","1","none")))).
lexitem("barátod",ln("friend","NOM",arg(f("sg","3"),s("sg","2","none")))).
lexitem("barátomat",ln("friend","ACC",arg(f("sg","3"),s("sg","1","none")))).
lexitem("barátjának",ln("friend","GEN",arg(f("sg","3"),s("sg","3","none")))).
```
d. Verbs:
```
lexitem("szeret",lf("love",feat("pres","indic"),arg(f("sg","3"),s("none","0","indef")))).
lexitem("szereti",lf("love",feat("pres","indic"),arg(f("sg","3"),s("none","0","def")))).
lexitem("szerettek",lf("love",feat("pres","indic"),arg(f("pl","2"),s("none","0","indef")))).
lexitem("szerettek",lf("love",feat("past","indic"),arg(f("pl","3"),s("none","0","indef")))).
lexitem("szeretnék",lf("love",feat("pres","cond"),arg(f("pl","3"),s("none","0","def")))).
lexitem("szeretnék",lf("love",feat("pres","cond"),arg(f("sg","1"),s("none","0","indef")))).
```
e. Participles:
```
lexitem("szerető",lif("love")).
lexitem("utáló",lif("hate")).
```

The parsing starts with the principal predicate **gramm_semantics**, which, after a successful morphosyntactic parsing carries out some semantic selection and gives a semantic representation formulated as a discourse-representation structure.

(10)
```
gramm_semantics(WL):-
    numberlist(1, WL, PWL), belongto(PWL, LIL2),
    finitelist(WL), norepart(WL),
    belonto2(LIL2, LIL, LIL2), satisfy(LIL, LIL2, GRALIST),
    semantics(LIL, SEMLIST, GRALIST, LIL),
    writeline(LIL), nl, writeline2(SEMLIST).
```

And there is a very similar predicate — **gramm_morphosyntax** — which differs only in one respect: it cannot provide semantic parsing; but it is necessary to separate semantic wellformedness from syntactic wellformedness.[11]

The input for the programme is a simple word list (variable WL), and as a first action the **numberlist** predicate gives serial numbers to words. This step seems to be technical but it is quite important because of unambiguous identification of the words in sentences. Then predicate **belongto** chooses, from the database section, the adequate lexical items for the words on the basis of *own words* which appear in the sentence (“in the surface structure”).

Predicate **belongto2** suggests grammatical relations on the basis of agreement and case relations observable in the sentence. This predicate refers to **relations** predicates, in which the environmental conditions of the given lexical items can be found.

[11] In the future it will be of great importance in translation to be able to parse sentences that are not grammatical according to semantics but grammatical according to morphosyntax (e.g. the sentence **The apple loves Mary*, where verb *love* has a semantic condition that the object must have a *+human* feature).

```
(11)  relations(N,SZERET,lf(X,F,A),[gr("subj","d",N,K),gr("subj","n",N,M),
      gr("obj","d",N,L),gr("obj","n",N,J)],LIL2):-
         subjd(Q,SZERET),in(li2(K,Q,LAB),LIL2),
         prec(li2(K,Q,LAB),li2(N,SZERET,lf(X,F,A)),LIL2),
         subjn(R,SZERET,LIL2),in(li2(M,R,LAB2),LIL2),
         prec(li2(M,R,LAB2),li2(N,SZERET,lf(X,F,A)),LIL2),
         objd(S,SZERET,LIL2),in(li2(L,S,LAB3),LIL2),
         prec(li2(N,SZERET,lf(X,F,A)),li2(L,S,LAB3),LIL2),
         objn(Z,SZERET,LIL2),in(li2(J,Z,LAB4),LIL2),
         prec(li2(N,SZERET,lf(X,F,A)),li2(J,Z,LAB4),LIL2),
         prec(li2(K,Q,LAB),li2(M,R,LAB2),LIL2),
         prec(li2(L,S,LAB3),li2(J,Z,LAB4),LIL2).
```

Variable N is the serial number of the lexical item, which plays an important role later. The following variables are the own word (from the sentence) and the label, and after that the most important component of this predicate comes, which is a list demonstrating the grammatical relations that the given lexical item demands. Because of unambiguity, these relations include the serial numbers mentioned above, and their formula is an ordered quadruple, e.g. `gr("subj","d", N, K)`, where the first string shows the name of the relation, the second string is the pillar, then the first number is the own serial number and the second number is that of the other word in the given relation: its environmental word.

In the example (11) shown above there is a transitive verb that has two arguments, a subject and an object, and four pillars — both arguments has a "nominal pillar" and a "determiner pillar." Between the verb and a pillar stands a grammatical relation that has conditions which are traced back to other predicates; see (12) below as an example:

```
(12)  subjd(X, Z):-
         ref(X), verb(Z).
      subjn(X, Z, LIL2):-
         in(li2(_,X,ln(_,"NOM",arg(Y,_))),LIL2),in(li2(_,Z,lf(_,_,arg(Y,_))),LIL2).
      objd(X, Z, LIL2):-
         in(li2(_,X,det(_,Y)),LIL2),in(li2(_,Z,lf(_,_,arg(_,s(_,_,Y)))),LIL2).
      objn(X, Z, LIL2):-
         in(li2(_, X, ln(_, "ACC", _)), LIL2), transitive(Z).
```

At this point the programme executes a "local search" — in the sense that every element is to find environmental words satisfying the appropriate grammatical relations. This is far from enough, however, because in this way a sequence of words like **Péter Mari alszik* 'Peter Mary sleeps,' for instance, would be predicted to be grammatical, because the verb can find a subject for itself and the two nouns can also find the verb for themselves. That is why some kind of mutual search is required, which means that members of a pair of words in a grammatical relation must find each other but no further words can be found for the same grammatical relation. The **`satisfy`** predicate in (10) carries out two important checking actions. First it has to consider the (quite simple) mutual search between lexical items: if there is a grammatical relation `gr("name", "pillar", N, M)`, there must be a relation with the same name and numbers `gr("name", "pillar", M, N))`. As a second operation, predicate **`immprec`** checks the right word order through the rank parameters ("weighing" the "intensity" of adjacency requirements between words in certain constructions; see (5)) in argument positions of predicate **`grimmprec.`**

```
(13)  grimmprec(7,"subjn",li2(A,_,_),li2(B,_,_),GRALIST):-
         in2(gr("subjn", A, B), GRALIST).
```

If all predicates are satisfied, the sentence is grammatical "according to" morphosyntax, and the last predicate writes the analysis. Let us demonstrate a running, where the question in the `goal` section is (14) below, and the answer is (15):

(14) `gramm_morphosyntax(["Péter","szeret","egy","barátságos","holland","lányt"]).`

(15)
```
["Péter","szeret","egy","barátságos","holland","lányt"]
li2(1,"Péter",ln("Peter","NOM",arg(f("sg","3"),s("none","0","none"))))
[gr("subj","d",1,2),gr("subj","n",1,2)]
li2(2,"szeret",lf("love",feat("pres","indic"),arg(f("sg","3"),s("none","0","indef"))))
[gr("subj","d",2,1),gr("subj","n",2,1),gr("obj","d",2,3),gr("obj","n",2,6)]
li2(3,"egy",det("a","indef"))
[gr("det","n",3,6),gr("obj","d",3,2)]
li2(4,"barátságos",adj("friendly"))
[gr("adj","none",4,6)]
li2(5,"holland",adj("Dutch"))
[gr("adj","none",5,6)]
li2(6,"lányt",ln("girl","ACC",arg(f("sg","3"),s("none","0","none"))))
[gr("det","n",6,3),gr("obj","n",6,2)]
```

4.2 Semantic representation. If a sentence has a right morphosyntactic output, predicate `semantics` carries out semantic selection, and if it also successful, it can provide the semantic representation: a DRS.

According to DRT, determiners (and proper names) provide referents, common nouns predicate something of them, and finite verbs provide a situation referent besides predicating something (of other referents). The following sentence in (16) below, for instance, is assigned the Prolog representation shown in (17) — collected in `SEMLIST` (10) —, which is practically the same as "box-representation" in DRT:

(16)	Péter szeret	egy	barátságos	holland	lány-t.
	Peter love-SG3	a	friendly	Dutch	girl-ACC
	'Peter loves a friendly Dutch girl.'				

(17)
```
[provideref("the",ref(1)), pred("Peter",[ref(1)])]
[provideref("state",ref(5)),
    pred("love",[ref(5),ref(1),ref(6)]),pred("past",[ref(5)])]
[provideref("a",ref(6))]
[pred("friendly",[ref(6)])]
[pred("Dutch",[ref(6)])]
[pred("girl",[ref(6)])]
```

r1, r2, r3
Peter(r1) friendly(r2) dutch(r2) girl(r2) love(r3, r1, r2)

In the semantic representation, determiners provide a referent in the following way:

(18) `sem(N,_,det(A,_),_,[`**`provideref(A,ref(N))`**`],_,_).`

During the creation of DRSs, the checked morphosyntactic output is used as an input. As it can be seen above in (18) the identifier of the provided referent can be the serial number that we used during the morphosyntactic parsing. The provided referent gets a parameter A which can take determiners 'the' or 'a' as its value (among others) to be given from the (input) label. As for nouns, sentence (16) above demonstrates the ground case, when they are DPs whose determiner provides a referent. In this case we use the grammatical relations shown below in (19) in order for the common noun to get its referent from its determiner ('K' below):

(19)
```
sem(N, _, ln(A, _, arg(f(_, "3"), s("none", "0", "none"))),
[gr("det", "n", N, K), _], [pred(A, [ref(K)])], _, _).
```

4.3 Possessive relations in Hungarian. In this subsection some practical parsing problems are demonstrated. Possessive relations in the Hungarian language raise a lot of parsing questions concerning morphosyntax and semantics, from agreement relations between possessions and possessors, through the "short" and "long" possessor, as far as multiple possessive relations.

The most interesting problems concern possessive relations containing determiners which do not appear in the (surface) sentence, as we insist on the basic principle of the GASG theory that in the morphosyntax only what we can "hear" can and should be described; i.e., no empty or deleted elements are used in the course of parsing.

It will be demonstrated what kind of tools can help us to solve these problems; for example, how our system can capture that the "short" possessor looks like an NP (with no determiner) in morphosyntax but its semantic representation is undoubtedly to include a DP. After all the interesting question for a linguist concerns the source of referents in DRSs when no suitable determiner can be found "in the surface."

During the morphosyntactic and semantic description the great variety of Hungarian possessive relations (see (20-23) below) requires a principled separation of different cases. Both in morphosyntax and in semantics the “short”, “long” and “possessed” possessors are to be handled separately.

(20) *“long” possessors*

a. a lány-nak a barát-ja
the girl-DAT the friend-POSS.3sg
‘the friend of the girl’

b. Péter-nek a barát-ja
Peter-DAT the friend-POSS.3sg
‘the friend of Peter’

(21) *“short” possessors*

a. a lány barát-ja
the girl friend-POSS.3sg
‘the girl’s friend’

b. Péter barát-ja
Peter friend-POSS.3sg
‘Peter’s friend’

(22) *the possessor is a missing personal pronoun*
a barát-om
the friend-POSS.1sg
‘my friend’ (or ‘the friend of mine’?)

(23) *multiple possessive relations*

a. a lány barát-já-nak a kutyá-ja
the girl friend-POSS.3sg-DAT the dog-POSS.3sg
‘the dog of the girl’s friend’

b. Péter barát-já-nak a kutyá-ja
Peter friend-POSS.3sg-DAT the dog-POSS.3sg
‘the dog of Peter’s friend’

c. a lány-nak a barát-já-nak a kutyá-ja
the girl-DAT the friend-POSS.3sg-DAT the dog-POSS.3sg
‘the dog of the friend of the girl’

d. Péter-nek a barát-já-nak a kutyá-ja
Peter-DAT the friend-POSS.3sg-DAT the dog-POSS.3sg
‘the dog of the friend of Peter’

By default, the possession searches the possessor as its argument, and it also searches the determiner pillar of this argument. However, this possessor can have a “long form” with a dative suffix *(-nAk)* or a “short form” (in the nominative case?) with no suffix. We would like to exhibit this difference by the names of the relations, too. Therefore the relation with a “short” possessor is denoted by `poss-n / poss-d`, and with a “long” possessor `gen-n / gen-d`. Two different lexical items may belong to the same own word in such cases, which can be distinguished in this way. For example, the own word *Péter* can be the subject or a “short” possessor in the sentence to be parsed. The database is to contain two lexical items with the same own word but different cases: *NOM(inative)* when the given word is analyzed as a subject and *POSS(essive)* in the latter case; see (24) below. What (25) demonstrates is that “long” possessors show a similar ambiguity between *DAT(ive)* and *GEN(itive):* the latter case can occur in possessive constructions.

(24)
```
lexitem("Péter",ln("Peter","NOM",arg(f("sg","3"),s("none","0","none"))))
lexitem("Péter",ln("Peter","POSS",arg(f("sg","3"),s("none","0","none"))))
```

(25)
```
lexitem("Péternek",ln("Peter","DAT",arg(f("sg","3"),s("none","0","none"))))
lexitem("Péternek",ln("Peter","GEN",arg(f("sg","3"),s("none","0","none"))))
```

The figures in (26-28) below show the relations in the possessive structures in (21)-(24).

(26) a. a lánynak a barátja — **det-n** (a – lánynak), **det-n** (a – barátja); **gen-d**, **gen-n**

b. Péternek a barátja — **det-n** (a – barátja); **gen-d+gen-n**

Grammatical relations of the possession in (26a) are a `det-n` relation with its own determiner, a `gen-d` relation with the deteminer pillar of its (“long”) possessor and a `gen-n` relation with the noun pillar of the possessor. When the possessor is a proper name (26b), the possession can find both pillars (`gen-d+gen-n`) in it (i.e. this proper name).

det-n

(27) a. a lány barátja b. Péter barátja

poss-n **poss-d+poss-n**

The case of "short" possessors (see (27a-b) above) is even more complicated, because in the surface structure short possessors appear with no determiner, and — according to the basic philosophy of GASG, mentioned above, that in the morphosyntax only what we can "hear" is to be described (independently of semantic circumstances) —, in these situations the possession cannot search a separate determiner pillar, it is to be regarded as lacking a `poss-d` relation, so it has only a `poss-n` relation with the noun pillar of the possessor, which means that the short possessor is practically an NP according to morphosyntax (27a); except for the case of proper names (27b), which can serve as the noun pillar and the determiner pillar at the same time.

(28) below demonstrates the intricate morphosyntactic relations in multiple possessive constructions:

det-n **det-n**

(28) a. a lány barátjának a kutyája

gen-d **poss-n** **gen-n**

det-n

b. Péter barátjának a kutyája

poss-d+poss-n **gen-d+gen-n**

det-n **det-n** **det-n**

c. a lánynak a barátjának a kutyája

gen-d **gen-n** **gen-n**

gen-d

det-n **det-n**

d. Péternek a barátjának a kutyája

gen-d+gen-n **gen-n**

gen-d

Let us turn to the question of semantic representation. The source of referents are determiners as a default. The finite verb also provides a (situation) referent and predicates something of other referents which are provided by determiner pillars — just as common nouns, which get their referent through its own determiner, and adjectives, which get their referent indirectly from the determiner pillar of the noun that they belong to.

In the case of possessive structures the starting-point is also that arguments (i.e. possessors) are DPs, so it has two pillars, when providing referents causes no special problem because determiners are available. Examples (26a,b), (27b), (28b-d) are representatives of this basic case. As for the type shown in (27a) and (28a), our proposal is based on "false NPs," which are arguments analyzed as NPs in the course of morphosyntactic parsing but regarded as DPs in semantics. How is it possible, i.e. what is the source of the determiner of the possessor when this argument is regarded as consisting of only the word *lány* 'girl'? The determiner pillar appears in the semantic characterization in the lexicon of the regent, that is, the possession *(barátja)*, and its type (its definiteness: `"an"` or `"the"`) is to be fixed as follows: it has to be the same as the type of the referent of the possession word. GASG supports this solution due to the fact that connections between morphosyntactic and semantic features are precisely fixed in the lexicon but the two levels of description need not be true reflections of each other at all.[12]

[12] There are also "true NPs" in Hungarian, where the referential type of a nominal expression is to be chracterized as "unspec" rather than "a" or "the": e.g. *vendég érkezett* 'guest arrived' in the sense that 'one or more guests have arrived.' The source of the determiner of the argument ('guest') in cases like this is also the lexical item of the regent ('arrived'). Predicative NPs (e.g. *Péter most* ***vendég*** 'Peter now guest' ('Now Peter is a guest.') mean another type of "true NPs"; where the common noun ('guest') needs a special lexical item in whose semantic component a referent is introduced.

5. Summary

A new sort of generative grammar, GASG, which is more consistently and radically lexicalist than any earlier generative grammar due to the fact that the lexicon plays the role of the generating component, has been demostrated in this paper.

Section 1 contains arguments in favor of the basic principle behind this enterprise, which can be called the Principle of Total Lexicalism. It also provides comments on the embedding of our grammar in the family of (non-transformational) generative grammars: GASG is to be regarded as a "radically lexicalist alternative" to Unificational Categorial Grammars, whose representative is Karttunen's CUG (1986), among others.

Section 2 provides a sketchy demonstration of GASG, especially the structure of (monostratal) lexical items, which play a central role in the theory in accordance with its lexical nature. Afterthat a separate section is devoted to the demonstration of rank parameters and the special way of satisfying lexical requirements on the basis of these parameters, because we regard this element of our grammar as promising a new chance to cope with stubborn problems of (free) word order.

Section 4 is about our first steps towards the Prolog implementation of GASG. In addition to grammaticality evaluation of sequences of words, our program can assign DRSs to grammatical sentences in a very quick and efficient, immediate, way (where 'immediate' means: not through sentence-level morphosyntactic representations). Practical parsing problems are discussed, too, in the intricate area of Hungarian possessive constructions (4.4), where "disappearing" articles should be accounted for by means of the totally lexicalist toolbox. The solution, here and elsewhere, is based on the characteristic property of GASG that connections between morphosyntactic and semantic features are precisely fixed in the lexicon but the two levels of description need not be true reflections of each other at all — that is why we can "afford" to insist on another basic principle of total lexicalism that in the morphosyntax only what we can "hear" can and should be described.

References

Alberti G. (1990), Quantifiers and the Chronology of Unification. Paper read at the *3rd Symposium on Logic and Language*, Révfülöp, Hungary. Unpublished.

Alberti G. (1996), Generative Argument Structure Grammar: A Strictly Compositional Syntax for DRS-Type Representations. Appeared in *Acta Linguistica Hungaria*, 1999.

Alberti G. (1997), Restrictions on the Degree of Referentiality of Arguments in Hungarian Sentences. *Acta Linguistica Hungarica* **44 (3-4).** 341-362.

Alberti G. (1998), GASG: The Grammar of Total Lexicalism Based on Prolog. To appear in *Studies in Applied Linguistics.* Univ. of Debrecen, Hungary (***deb98.doc).[13]

Alberti G. (1999), GASG: The Grammar of Total Lexicalism. In *Working Papers in the Theory of Grammar* **6/1,** Theoretical Linguistics Programme, Budapest University (ELTE) and Research Institute for Linguistics, Hungarian Academy of Sciences, Hungary (***gasg98.doc).

Alberti G. (2000a), Lifelong Discourse Representation Structures. *Gothenburg Papers in Computational Linguistics* 00-5, 13-20 (***gotapite.pdf).

Alberti G. (2000b), Indo-Germanic Word Order Phenomena in a Totally Lexicalist Grammar. To appear in *Sprachtheorie und germanistische Linguistik.* Kossuth Egyetemi Kiadó, Debrecen, Hungary (***kertsz00.doc).

Alberti G. (2001), Morphological Relations and Idioms in a Totally Lexicalist Grammar. In I. Kenesei (ed.) *Argument Structure in Hungarian*. Akadémiai Kiadó, Budapest, Hungary, 201-235 (***kenang00.doc).

Alberti G. (2002), Távolsági kapcsolatok kezelése egy totálisan lexikális grammatikában [How to Treat Long Distance Dependencies in a Totally Lexicalist Grammar], Textologica **6**, Debrecen, Hungary.

Chomsky, N. (1957), *Syntactic Structures.* The Hague, Mouton.

Chomsky, N. (1995, ed.), *The Minimalist Program.* MIT Press, Cambridge, Mass.

[13] ***: http://maya.btk.pte.hu/~anita/gelexi/… (here and later in the References)

Dowty, D. R. (1996), Toward a minimalist theory of syntactic structure. In H. Bunt and A. van Horck (eds.), *Discontinuous Constituency*. Mouton de Gruyter, Berlin & New York.

Kamp, H. (1981), A theory of truth and semantic representation. In Groenendijk, J., Janssen, T., and Stokhof, M. (eds.). *Formal methods in the study of language*. Amsterdam: Mathematical Centre.

Kamp, H., and U. Reyle (1993), *From Discourse to Logic.* Kluwer Academic Publ.

Karttunen L. (1986), *Radical Lexicalism.* Report No. CSLI 86-68. Stanford.

Lehmann, Ch. (1988), On the Function of Agreement. In H. Barlow, Ch. A. Ferguson (eds.), *Agreement in Natural Languages. Approaches, Theories, Descriptions.* CSLI Stanford.

Marantz, A. (1995), The Minimalist Program. In G. Webelhuth (ed.), *Government and Binding Theory and the Minimalist Program.* Blackwell, Oxford & Cambridge. 349-382.

Partee, B. H., Meulen, G. B. ter, and Wall, R. P. (1990), *Mathematical Methods in Linguistics.* Kluwer Academic Publ.

Shieber, S. M. (1985), Evidence against the context-freeness of natural languages. *Linguistics & Philosophy* **8**, 333-343.

van Eijck, J., and H. Kamp (1997), Representing discourse in context. In J. van Benthem, A. ter Meulen (eds.): *Handbook of Logic and Language.* Elsevier, Amsterdam, and MIT Press, Cambridge, Mass.

Zeevat, H. (1991a), A Compositional Version of Discourse Representation Theory. In *Linguistics and Philosophy* **12**, 95-131.

Zeevat, H. (1991b), Aspects of Discourse Semantics and Unification Grammar. Diss., Univ. of Amsterdam