Total Lexicalism and GASGrammars: A Direct Way to Semantics

G. Alberti - K. Balogh - J. Kleiber - A. Viszket

University of Pécs
Department of Linguistics

Abstract. A new sort of generative grammar (Sec2) will be demonstrated which is more radically "lexicalist" than any earlier one (Sec1). It is a modified Unification Categorial Grammar [1–4] from which even the principal syntactic "weapon" of CGs, Function Application, has been omitted. What has remained is lexical sign and the mere technique of unification as the engine of combining signs. The computation thus requires no usual linguistic technique (e.g. Move, Merge, traces [5], Function Application [6]); which promises a straightforward implementation of GASG in Prolog. Our parser decides whether a Hungarian sentence is grammatical and creates its (practically English) DRS (Sec3).

1 DRT, UCG and Total Lexicalism

A "totally lexicalist" generative grammar will be demonstrated in this paper. The first motivation of the enterprise was the stubborn problem of compositionality in DRT (Discourse Representation Theory; e.g. [7], [4]).

DRT is a successful attempt to extend the sentence-level Montagovian model-theoretic semantics to the discourse level. Its crucial proposal is that a level of discourse representation *must* be inserted in between the language to be interpreted and the world model serving as the context of interpretation. The insertion of this level, however, has given rise to a double problem of *compositionality* (language → DRS, DRS → world model), at least according to the very strict sense of the Fregean principle of compositionality introduced by Montague [8]. As for the DRS → world model transition Zeevat [2] has provided a compositional solution, which could successfully be built in the new version of DRT [4]. As for the language → DRS transition, however, the authors admit (p195) that no (properly) compositional solution could be found in the last two decades.

The failure of elaborating a properly compositional solution to the language → DRS transition arises from the fundamental incompatibility of the strictly hierarchically organized generative syntactic phrase structures (PS; e.g. [9], [5]) with the basically unordered DRSs. Nowadays [2], [4] some kind of Categorial Grammar (CG) is held to promise the best chance for capturing the language → DRS transition in a properly compositional manner. The reason lies in the fact that, in a CG system, language-specific information (about how words can combine to form constituents, and then sentences), stored in PS rules in the transformational generative theory, is stored in the Lexicon; the reduced syntax only "concatenates": it permits the words with compatible lexical information to combine (this operation of concatenation is referred to as *Function Application)*. The problem with Classical CG is that it has only a context free generative capacity, which is held 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 or to introduce the technique of *unification*, applied e.g. in Prolog (UCG). It is straightforward in the spirit of what has been said so far that DRT is (more) compatible with UCG insisting on a reduced syntax.

UCG is a monostratal grammar, which is based on the formalized notion of the Saussurean sign: a structure that collects a number of levels of linguistic description and expresses relations between the levels by sharing variables in the description of the level information [3 : p145]. The set of well-formed expressions is defined by specifying a number of such signs in the lexicon and by closing them under rule applications (i.e. the selected lexical signs can be combined to form sentences via a finite number of rule applications). In monostratal grammars the syntactic and semantic operations are just aspects of the same operation. A prime example of such grammars, besides UCG, is HPSG.

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 [1 : p19] 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." Let us take this latter remark on trees and feature sets seriously: adjacency of words is to be registered in the course of analysis exclusively and precisely in the linguistically significant cases. The corresponding technique is to be based on an approach where adjacency and order among words are treated by, instead of the usual categorial apparatus, the same technique of unification as morphological cohesion. And what will be then the "engine" combining words to form sentences (since in CGs the lexical features of words only serve as filters to avoid inappropriate combinations)?

There is no need for a separate engine at all! The engine must be unification itself, which is capable of running Prolog programs properly. The rich description of a lexical sign 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 finding 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 at our disposal.

Sec2 provides the system of definitions of generative argument structure grammars, and in the last section our parser is sketched.

2 Definition System of GASGrammars

First of all, we provide the abstract definition of *language*, which is similar to the one in [6]. Different alphabets (e.g. that of sounds and intonational symbols) can be considered, however, depending on the task, and the definition of phonological model is ambitious: it is the (morpho-) phonologist's task to collect both the relevant set of morpheme segments and the relations among them[1].

[**Def1:** 1.1. Let A be a finite set: the alphabet. Let # and "." are special symbols which are no members of A: the *space* symbol and the *full stop*. Suppose that, together with other symbols, they constitute a set Y, that of *auxiliary symbols.* A member s of $(A \cup Y)^*$ is called a *sentence* if at least one of its members is an element of A, $(s)_1 \neq \#, (s)_1^R = .$, there are no further full stops in the list, and $(s)_i = \# = (s)_{i+1}$ for no i.
1.2. An element of A^* is the *i-th word* of a sentence s if it is the affix of s between the $i-1$-th and the i-th symbol #; the *first word* is the prefix of s before the first #, and if the last # is the j-th, the suffix of s after it (and before the full stop) is the $j+1$-*th*, or *last, word.*
1.3. We call a subset L of $(A \cup Y)^*$ a *language (over alphabet A)* if all of its members are sentences.
1.4. We call Phon = ⟨Mors, Rel⟩ a *phonological model (over alphabet A)* if Mors is a subset of A^*, called a set of *morpheme segments*, and Rel is a set of relations in Mors.]

Numbering will prove to be a crucial question because corresponding elements of intricately related huge representations should be referred to.

[**Def2:** 2.1. Let s be a sentence of a language L over an alphabet A. We call an element n of $(\mathbf{N}^3)^*$ a *(three-dimensional) numbering* if $(n)_1 = \langle 1,1,1\rangle$, [if $(n)_m = \langle i,j,k\rangle$, either the first projection of $(n)_{m+1}$ is i or $(n)_{m+1} = \langle i{+}1,1,1\rangle$], and [for each number i in the first projection, the set of second elements consists of natural numbers from 1 to a maximal value p, and for each pair $\langle i,j\rangle$ there are exactly the following three members in the numbering: $\langle i,j,1\rangle, \langle i,j,2\rangle$ and $\langle i,j,3\rangle$, necessarily in this order (but not necessarily next to each other)].
2.2. An element mos of $(\mathbf{N}^3 \times A^*)^*$ is a morphological segmentation of s if [the

[1] In Hungarian, for instance, but not in English, the following relations necessarily belong to Rel: [the morpheme segment in question consists of a single vowel which is j or it is empty], [. . . is *a, á, e* or *é*]. We assume that the 3sg. possessive morpheme consists of such two segments (and a third empty one): e.g. *szárny-a-ként* 'wing-poss3sg-as' (as its wing), *szárny-á-t* 'wing-poss3sg-ACC,' *fej-e-ként* 'head-poss3sg-as,' *fej-é-t* 'head-poss3sg-ACC,' *part-ja-ként* 'beach-poss3sg-as,' *part-já-t* 'beach-poss3sg-ACC,' *medvé-je-ként* 'bear-poss3sg-as,' *medvé-jé-t* 'bear-poss3sg-ACC.'

$[1,2,3]$-projection of mos is a numbering (the numbering of mos)], [it is excluded in the case of each pair $\langle i,j\rangle$ that all three fourth members belonging to the triples $\langle i,j,1\rangle, \langle i,j,2\rangle$ and $\langle i,j,3\rangle$ in mos are empty lists], and [for each number u of the domain of the first projection of mos, the $u-$th word of s coincides with the concatenation of the fourth projection of the element of mos of the form $\langle u,1,1,\_\rangle$ with the fourth projections of all the following elements with number u as its first projection, just in the order in mos].
2.3. If $\langle i,j,k,a\rangle$ is an element of mos, we say that a is the $\langle i,j,k\rangle$-th morph segment of the given morphological segmentation; we can also say that the triple consisting of the $\langle i,j,1\rangle$-st, $\langle i,j,2\rangle$-nd and $\langle i,j,3\rangle$-rd morph segments, respectively, is the $\langle i,j\rangle$-th morph of mos.]

Thus each morpheme is divided into exactly three segments, $\langle i,j,1\rangle, \langle i,j,2\rangle$ and $\langle i,j,3\rangle$ (out of which at most two are allowed to be empty). Why? In Semitic languages certain stems are discontinuous units consisting of three segments between which other morphemes are to be inserted in. It is allowed in GASG that the cohesion between a morpheme and a particular segment of another morpheme is stronger that the cohesion between the three segments of the latter.

In Hungarian, segments of the same morpheme can never be separated from each other. It is useful, however, to refer to a certain segment of a morpheme — in cases where another morpheme determines just the segment in question[2]. Segmentation into just three parts is proposed as a language universal.

Important numbering techniques are defined below again.

[**Def3:** 3.1. We call an element m of $(\mathbf{N}^2)^*$ a strict *(two-dimensional) numbering* if $(m)_1 = \langle 1,1\rangle$, and [if $(m)_k = \langle i,j\rangle$, then $(m)_{k+1} = \langle i,j+1\rangle$ or $\langle i+1,1\rangle$].
3.2. A two-dimensional numbering m is a *homomorphic correspondent of* a three-dimensional numbering n if there is a function hom such that for each triple $\langle i,j,k\rangle (k = 1,2,3)$ occurring in n, $\mathrm{hom}(\langle i,j,k\rangle) = \langle i,j\rangle$; which can be said as follows: member $\langle i,j,k\rangle$ of the three-dimensional numbering is the *k-th segment of* member $\langle i,j\rangle$ of the two-dimensional numbering.]

Despite their great length, Def4-6 are worth commenting together because the intricate construction of gasg's described in Def4 can be evaluated through understanding its functioning: generation (acceptance) of sentences.

[**Def4:** 4.1. A sextuple $G = \langle A, Phon, B, int, X, R\rangle$ is a *generative argument structure grammar (gasg) belonging to the phonological model* Phon $= \langle$Mors, Rel$\rangle$ over alphabet A (see def1.4.) if [X is a list of lexical items [def4.3.] whose members are elements of Lex(Term)], and [R is a ranked rule system [def4.4.] also over term set B [def4.2.].
4.2. B, the set of *basic terms*, is the sequence of the following sets:

$Con(j) = \bigcup Con(j)_i$, for $j = 1,2,31,32$, and $i = 0,1,2,\ldots$: finite sets of *constants of arity i,*

[2] In this footnote Hungarian morphs are demonstrated with stable first and third segments but altering middle ones: *al-hat* ‘sleep-may,’ *szúr-hat* ‘prick-may,’ *kér-het* ‘ask-can,’ *űz-het* ‘chase-can.’ Besides this frontness harmony, other morphemes are sensitive to roundness as well.

$Icon(j) = \bigcup Icon(j)_i$, for$j = 1, 2$, and $i = 1, 2,$: finite sets of *interpretable constants of arity* i*;* int can be defined here as a total function int: $Icon(j) \rightarrow$ Rel,
Numb: a set of *numbers* that necessarily includes all natural numbers,
VAR_0: variables that can substitute for elements of $Con(2)_0$ and Numb,
Rank $= \{r_1, \ldots, r_K\}$ (K=7).

4.3. A lexical item is a triple li = ⟨ownc, frmc, pdrs⟩ where [1–3]:

1. Set ownc, *own word conditions*, is a subset of the following set Form(1) of well-formed fomulas:
   (a) For an arbitrary $p \in Icon(1)_k, k = 1or2$, the expression $p(t_1, \ldots, t_k) \in Form(1)$ where an argument t_i is a term, $i = 1, 2, \ldots, k$.
   (b) Triples of numbers, precisely, elements of $Numb^2 \times \{1, 2, 3\}$ are terms; and lists of terms are also terms.
   (c) Formula $p \vee q$ is an element of Form(1) if p and q are its elements.
2. Set frmc, *formal conditions*, is a subset of the following set Form(2) of well-formed fomulas:
   (a) For an arbitrary $p \in Con(2)_k, k = 2, 3, \ldots$, the expression $p(t_1, \ldots, t_k) \in$ Form(2) where argument t_i is a term for $i = 2, \ldots, k$, but $t_i \notin$ Rank for these values of i, whereas $t_1 \in$ Rank. We call the formulas defined in this step *ranked formulas.* We also say that out of these ranked formulas those which are members of set frmc *belong to* the given lexical item li.
   (b) For an arbitrary $p \in Con(2)_k$ or $p \in Icon(2)_k, k = 1, 2, \ldots$, the expression $p(t_1, \ldots, t_k) \in$ Form(2) where argument $t)i$ is a term for $i = 1, 2, \ldots, k$, but $t_i \notin$ Rank for these values of i.
   (c) Elements of $\bigcup Con(2)_i$, for $i = 0, 1, 2, \ldots$, are terms;
   elements of $\bigcup Icon(2)_i$, for $i = 0, 1, 2, \ldots$, are terms;
   elements of Numb and VAR$_0$ are terms;
   lists of terms are also terms;
   elements of Form(2) which are not ranked formulas are all terms too.
   (d) Formula $p \vee q$ is an element of Form(2) if p and q are its elements.
   (e) Formula $p \vee q$ is an element of Form(2) if p and q are its elements.
3. Set pdrs, *the proto-DRS provided by the given lexical item*, is a pair ⟨ bdrs, embc ⟩ where bdrs (the *basic* DRS) is a subset of the following set Form(31) of well-formed fomulas, and embc (the *embedding conditions)* is a subset of set Form(32) of well-formed fomulas defined after that:

   (a) For an arbitrary $p \in Con(31)_k$, the expression $p(t_1, \ldots, t_k) \in$ Form(31) where an argument t_i is a term. If the given formula is an element of subset bdrs, the terms occupying its argment positions are called *referents belonging to* bdrs.
   (b) Elements of $\{ref\} \times (Numb\, VAR_0)^3$ are terms where ref is a distinguished element of Con(31)$_3$.
   (c) The expression $p(t_1, \ldots, t_k) \in$ Form(32) where argument t_i is a term for $i = 1, 2, \ldots, k$, and $p \in$ {oldref, newref} = Con(32)$_1$ or $p \in$ {fixpoint, $\langle, \leq, \neq, \sim\} = Con(32)_2$.

(d) Elements of $\{ref\} \times (Numb\, VAR_0)^3$ are terms where ref is a distinguished element of $\mathrm{Con}(32)_3$, and it is also a restriction that a quadruple $\mathrm{ref}(i, j, k)$ can be considered here a term if it is a referent belonging to set bdrs.

4.4. The *ranked rule system* denoted by R is defined as an arbitrary subset of the set rr(Form(2)) of *ranked rules over* set Form(2) of formulas (defined in def4.3.2.): all formulas of the form $p \leftarrow q$ is an element of rr(Form(2) if p is a ranked formula, and [q is a conjunction of elements of Form(2): $q = q_1 \wedge q_2 \wedge \ldots \wedge q_d$ for some d].]

[**Def5:** 5.1. An element num of $(\mathbf{N}^2 \times X)^*$ is called a *numeration* (over a gasg G) if [the [1,2]-projection of the list is a strict two-dimensional numbering], and [members of the third projection are lexical items (coming from the fifth component of G)].
5.2. If $\langle i, j, li\rangle$ is an element of num, we can say that the given lexical item li is the $\langle i, j\rangle$*-th element* of the numeration.]

[**Def6:** 6.1. A sentence s – a member of $(A \cup Y)^*$ in Def1, is *grammatical* according to a gasg $G = \langle A,\mathrm{Phon}, B,\mathrm{int}, X, R\rangle$ if

> there is a numeration num of $(\mathbf{N}^2 \times X)^*$,
> there is a *(cohesion)* function coh: $\mathrm{VAR}_0 \rightarrow Con(2)_0 \cup Numb$ (def4.2.!),
> and sentence s has a morphological segmentation mos of $(\mathbf{N}^3 \times A^*)^*$ (Def2.2.) such that the numbering of numeration num is a homomorphic correspondent of the numbering of segmentation mos
> and the $\langle$coh,int$\rangle$ pair satisfies [def6.2.] numeration num according to rule system R.

6.2. Pair $\langle$coh,int$\rangle$ *satisfies* (def6.2.) numeration num according to rule system R if for each possible $\langle i, j\rangle$, the lexical item li which is the $\langle i, j\rangle$-th member of the numeration is satisfied. This lexical item li = $\langle$ownc,frmc,pdrs$\rangle$ is *satisfied* if its all three components are *satisfied*.

1. Formula set ownc is *satisfied* if,
   [in the case of 4.3.1.a., $\langle \mathrm{int}'(t_1), \ldots ,\mathrm{int}'(t_k) \in \mathrm{int}(p) \in \mathrm{Rel}$, where (Rel is the set of relations in the phonological model Phon belonging to gasg G, and) function int′ is an extension of int that assigns a number triple $\langle i, j, k\rangle$ the $\langle i, j, k\rangle$-th morph segment of the morphological segmentation mos, and a number pair $\langle i, j\rangle$ the $\langle i, j, 1\rangle$-st morph of mos];
   [in the case of 4.3.1.c., p is satisfied or q is satisfied].
2. Formula set frmc is *satisfied* if one of the cases discussed below is satisfied. First of all, however, coh'(p) is to defined for elements of formulas of Form(2) and Form(3): it is a formula whose only difference relative to p is that each occurrences of variable v (elements of VAR_0) has been replaced with coh(v).
   In the case of 4.3.2.a., a ranked formula $p(t_1, \ldots , t_k)$ is satisfied if there is a formula $p(t'_1, \ldots , t'_k) \leftarrow q'$ in rule system R such that

coh($p(t'_1,\dots,t'_k)$) $= p(t_1,\dots,t_k)$, there is a formula q such that coh(q) $=$ coh(q'), and q belongs to the $\langle i',j'\rangle$-th lexical item in numeration num for an arbitrary pair $\langle i',j'\rangle$, and coh(q') is satisfied.

In the case of 4.3.2.b., a formula $p(t_1,\dots,t_k)$ is satisfied if

EITHER there is a formula $p(t'_1,\dots,t'_k) \leftarrow q'$ in rule system R such that coh($p(t'_1,\dots,t'_k)$) =coh($p(t_1,\dots,t_k)$), there is a formula q such that coh(q) $=$ coh(q'), and q belongs to the $\langle i',j'\rangle$-th lexical item in numeration num for an arbitrary pair $\langle i',j'\rangle$, and coh(q') is satisfied *(indirect satisfaction),*

OR *coh*($p(t'_1,\dots,t'_k)$) belongs to the $\langle i',j'\rangle$-th lexical item in numeration num for an arbitrary pair $\langle i',j'\rangle$ *(direct satisfaction),*

OR $\langle$int$'$(coh(t_1)),...,int$'$(coh(t_k))$\rangle$ $\in$int(p) $\in$Rel (int$'$ has been defined in def6.2.1. *(direct satisfaction).*

In the case of 4.3.2.d., $p \vee q$ is satisfied if p is satisfied or q is satisfied.

In the case of 4.3.2.e., $p \wedge q$ is satisfied if p is satisfied and q is satisfied.

3. Formula sets bdrs and embc are satisfied if each formula p that can be found in one of them is satisfied. This arbitrary formula p is *satisfied* without conditions.

6.3. Let us denote sem the set consisting of the $\langle$coh(bdrs),coh(embc)$\rangle$ for all *lexical items* in the numeration. We can call it the *discourse-semantic representation of sentence s.*]

In harmony with our "total lexicalism," lexical item is the crucial means of a gasg (def4.3.). Its first component out of the three (def4.3.1.) consists of conditions on the *"own word"* deciding whether a morpheme in a (potential) sentence can be considered to be a realization of the given lexical item (see def6.2.1. and the last two footnotes on allomorphs). It is our new proposal [12] that, instead of fully inflected words (located in a multiple inheritance network), li's are assigned to morphemes – realizing a "totally lexicalist morphology"

The component of formal conditions (def4.3.2.) is responsible for selecting the other li's with which the li in question can stand in certain grammatical relations (def6.2.2.). It imposes requirements on them and exhibits its own properties to them. As for the range of grammatical relations in a universal perspective [10], there are unidirectional relations, e.g. an adjective "seeks" its noun, where the "seeking" li may show certain properties (number, gender, case, definiteness) of the "sought" one, and bidirectional relations, e.g. an object and its regent (in whose argument structure the former is) "seek" each other, where the argument may have a case-marking depending on the regent, and the regent may show certain properties (number, person, gender, definiteness) of the argument. The rule system in the sixth component of gasg's (def4.4.), among others, makes it possible to store the above listed language-specific factors outside li's so $frmc$ (def4.3.2.) is to contain only references to the relations themselves.

It is *ranked* implication rules (def4.3.2., def6.2.2.) that we consider to be peculiar to GASG. In addition to satisfying a requirement described in a li *directly* by proving that either some property of another li is appropriate or the morphemes / words in the segmented sentence stand in a suitable configuration, the requirement in question can be satisfied *indirectly* by proving that there is a lexical item which has a competitive requirement ranked higher. This optimalistic technique enables us to dispense with phrase structure rules: the essence (precise details in [13,14]) is that, if word (morpheme) w_1 stands in a certain relation with w_2, w_1 is required to be adjacent to w_2, which can be satisfied, of course, by putting them next to each other in a sentence, but we can have recourse to an indirect way of satisfaction by inserting other words between them whose adjacency requirements (concerning either w_1 or w_2) are ranked higher (and these intervening words, in a language-specific way, may be allowed to "bring" their dependents). In def4.2. *seven* ranks are proposed as a universal concerning the complexity of human languages.

The discourse-semantic component of li's (def4.3.3.) is practically intended to check nothing[3] (def6.2.3.) but their "sum" coming from the whole numeration (def6.3.) provides a "proto-" DRS in the case of sentences that have proved to be grammatical. Our proto-DRSs seem to have a very simple structure in comparison to DRSs with the multiply embedded box constructions demonstrated in [11]. Nevertheless, they store the same information due to the conditions of a special status defined in def4.3.3.2. Moreover, several cases of ambiguities can simply be traced back to an underspecified state of these special conditions. Let us consider an illustration of these facilities.

(1) Most widowers court a blonde.

(2)

most($e_0; e_1, e_2$)	fixpoint(e_0), $e_0 < e_1$, $e_1 < e_2$, newref(e_0)
widower($e_1; r_2$)	newref(e_1), newref(e_2)
court($e_2; r_2, r_3$)	newref(r_2), $e_1 \approx r_2$
blonde(r_3)	newref(r_3), $r_3 \approx$???

(3) $e_2 \approx r_3$: 'It is often true that if someone is a widower he courts a blonde.'
$e_0 \approx r_3$: 'There is a blonde whom most widowers court.'

The basic proposition (whose eventuality referent is e_0) is that a situation [e_1: somebody is a widower] often implies another situation [e_2: he courts somebody]; symbols '<' refer to these situations' not being facts but their and some of their characters' belonging to fictive worlds [15]. The widower necessarily belongs to the fictive world of our thinking about an abstract situation ($e_1 \approx r_2$). But which world does the blonde belong to? Referent r_3 is looking for its place... And it can find its place in different worlds (3) – without assuming different syntactic structures behind the two readings[4].

Let us finish the section with the definition of a *language generated by a gasg:*

[**Def7:** In the circumstances defined above in def6, we can say that gasg G *generates* sentence s *through* segmentation mos and numeration num, and G *assigns*

[3] Semantic restrictions (e.g. on the [+human] status of an argument) can be put in the set of formal conditions (def4.3.2.) among morphologic and syntactic ones.

[4] The freedom in finding the appropriate world has language-dependent restrictions depending also on the argument status and other grammatical relations of the li of the indefinite article in question, of course.

the given sentence DRS sem as its *discourse-semantic representation.* It can also be said in this situation that gasg G has generated reading $\langle s, mos, num, sem\rangle$. $L(G) \subset (A \cup Y)^*$ is called the language defined by gasg G if $L(G)$ consists of the sentences generated by G.]

3 Implementation in Prolog

Our work is permanently developed, and the version which is available now can parse uncompound neutral Hungarian sentences. In our parser we insist on the theoretically clear principles of GASG, but naturally we have to make some technical changes according to the special features of programming in Prolog. Hence, parts of the lexical items in GASG are stored in different places in the program. The `database` section contains the lexical items which are morphemes and consist of the ownword, phonological features and some inherent syntactic conditions (e.g. the argument structure). Other environmental conditions and properties of morphemes that a lexical item searches are put down in the `synrelations` predicate. This part means the syntactic parsing together with a checking that contains the `immprec` predicate. The third part of a GASG lexical item – which is semantics – is represented in the `semantics` predicates.

The parsing starts with the main predicate `gramm`, which, after a successful phonological and morphosyntactic parsing, gives semantic representation formulated as a DRS:

```
gramm(SENTENCE):-
   words(SENTENCE,WL1), corr(WL1,WL), morphwl(WL, MLABL),
   numberlist(1,MLABL,NMLABL), phon(NMLABL,WL), immprec(NMLABL),
   synrel(NMLABL, SYNRELLIST, MIXEDLIST),
   semantics(MIXEDLIST, DRS, SYNRELLIST, MIXEDLIST),
   write(S), nl, writeline(NMLABL), nl,
   writeln2(SYNRELLIST), nl, writeln3(DRS).
```

The first six predicates provide for the morphophonological cheking. The input is a simple string e.g.: `"A fiú beül a székbe."` ‘the boy in-sit the chair-INESS’ (The boy sits into the chair.). The `words` and the `corr` predicates find the words in the string and give us a list: `["a", "fiú", "beül", "a", "székbe"]`, and after this the `morphwl` predicates search the morphemes in the sentence according to the lexical items in the database section. Before the linguistic parsing there is a technical but quite important step: to give serial numbers to the morphemes. It is necessary because of the unambiguous identification of the morphemes in the sentence. The morphemes get double numbers that shows in which word is which morpheme. For example in the sentence *Péter be-ül-tet-i a lány-t a szék-be* ‘Peter in-sit-cause-3sg.defobj the girl-ACC the chair-INESS’ (Peter sits the girl into the chair.) the morpheme *-tet* gets the numbers (2,3). In this way we can always refer squarely to the morphemes.

The `database` section contains such lexical items as it is shown below:

```
   lexi(m("","be",""),labder("into",phonfsu(1,2,-1,2),2,prt1("ILLAT"))).
   lexi(m("","ül",""),labstem("sit",phonfst(1,1,1,2),2,[["NOM","LOC"]])).
```

```
lexi(m("t","A","t"),labder("cause",phonfsu(2,2,0.2,2),2,ac(-1,0,1))).
```

All lexical items contain the ownword of the morpheme (`m("","fiú","")`), and a "label" with the English "translation", the phonological features (`phonfst`), the category (1=noun/suffix for nouns, 2=verb/suffix for verbs, 3=determiners, 4=adjunct) and the syntactic conditions.

In this phase of the programme we can already account for such phonological phenomena as vowel-harmony, lowering, V∼ ⊘ alternation, linking vowels, lengthening, shortening etc. Phonologically two kinds of requirements are needed. The first one accounts for the choice of the possible realizations of the given morpheme (lexical item), these possible realizations are technically variables in the own words. E.g. in the case of *bokor* ('bush') the own word is *bokOr*, and the phonological realization depends on the following suffix: *bokor-ban* 'bush-INESS' but *bokr-ot* 'bush-ACC'. Or in the case of the suffix *-ban/-ben* ('in') the own word is *-bAn*, and the frontness of the vowel depends on the frontness of the stem: *bokor-ban* 'bush-INESS' *but szék-ben* 'chair-INESS'. The other kind of requirements is how the lexical items effect on the phonological realizations of other lexical items in the same word (e.g. lowering stems or suffixes, or again vowel-harmony).

The most simple example for indirect satisfaction is the calculation of order of morphemes within words. Every suffix would like to be adjacent to the stem, but these requirements are not equally strong. According to the definition, if a requirement cannot be satisfied directly (there are more than one suffix in a word), it could be satisfied indirectly. If a suffix A wants to be adjacent to the stem on rank α, and a suffix B wants to be adjacent to the stem on rank β, and $\alpha < \beta$ then the morpheme order is: stem, A, B.

The checking/parsing demonstrated above gives us a list which calls the synrel predicate, which provides the syntactic parsing accordig to the morphemes in the words of the sentence. The synrel predicate calls the synrelations predicates, namely the morphemes call their own syntactic requirements. In this way the program creates a new list, where next to the morphemes there is always another list, which contains the grammatical relations that the given morpheme can establish in the given sentence. The representation of a grammatical relation is an ordered septuple: `gr[X,Z,Y, N,M, K,L]`. In the expression the first three elements are the determiners of the relation: the first string is the name of the element that calls the relation, the second string is the environmental element that the first one searchs and the third one is the type of the relation. The other four elements in the representation are the numbers of the morphemes that have the relations.

In our system finite verbs look for the two pillars of their arguments – the arguments are defined in the lexical item. For example a non-transitive verb searchs the noun pillar and the determiner pillar of its nominativ argument (relations: `gr("regent", "noun", "subj", X, Y, N, M)` and `gr("regent", "det", "subj", X, Y, K, L)` and a transitive verb searchs four elements: the noun and determiner pillar of its nominative argument (the same as before) and looks for the determiner pillar and an accusative suffix as the representative of the noun pillar of its accusative argument. Determiners look for a noun stem for relation `gr("det", "noun", "free", X, Y, K, L)` and the stem of the finite verb for relation `gr("det", "regent", "_", X, Y, K, L)`. The common nouns search the finite verb for a *subject* relation if they do not have a case marking suffix. In the case when the noun has a case marking suffix, it looks for the environmental morpheme. And finally the affixes search the stem for `gr("pref/suff", "stem", "free", X, Y, N, M)` and an environmental morpheme for a grammatical relation. For example the prefix *ba-* ‘in’ searchs a case marking suffix, which is the *-bAn* ‘INESS’.

At this point the program executes a “local search” – in the sense that every morpheme is to find environmental morphemes satisfying the appropriate grammatical relations. But this is far from enough becuse in this way sentence *A fiú a lány ül* ‘The boy the girl is sitting’ could be accepted as a grammatical one. That is why some mutual search is required, which means that members of a pair of morphemes in a grammatical relation must find each other but no further morphemes can be found for the same relation. The mutual search is satisfied if every relation `gr(A,B,REL,X,_,Z,_)` finds the relation `gr(B,A,REL,Z,_,X,_)`.

If all predicates above are satisfied, the sentence is grammatical “according to” morphosyntax, and the program gives us a right morphosyntactic output, which calls the predicate `semantics`.

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

According to DRT, determiners (and proper names) provides referents, common nouns predicate something of them, and finite verbs provide a situation referent besides predicating something (of other predicates). In our new conception determiners tell in which world they provide the given referent [15]. The output of our semantic representation is shown in (4-5). The referents contain three numbers that refer to the morpheme that has provided it (e.g. r(3,1,1)=the first provided referent by the first morpheme of the third word). The ordering between the worlds they belong to (see (2-3)) are also represented by the following relations: ∼, <or=, <.

(4) A fiú be-ül-tet-het-i a büszke medvé-jé-t a szék-em-be.
the boy in-sit-cause-may-sg3.objdef the proud bear-poss.3sg-ACC the chair-poss.1sg-INESS
‘The boy can sit his/her proud bear in my chair.’

(5) semantic output for sentence (4):

```
provref("old",[r(1,1,1)])
provref("<or=",[r(1,1,1),(e(4,4,1)])
pred("clever",[r(1,1,1)])
pred("boy",[r(1,1,1)])
provref("new",[e(4,2,1)])
provref("∼",[(e(4,3,1),e(4,2,1)])
pred("sit_into",[e(4,2,1),r(5,1,1),r(8,1,1)])
provref("new",[e(4,3,1)])
provref("<",[(e(4,4,1),e(4,3,1)])
pred("cause",[e(4,3,1),r(1,1,1),e(4,2,1)])
provref("fixpoint",[e(4,4,1)])
pred("may",[e(4,4,1),r(1,1,1),e(4,3,1)])
provref("old",[r(5,1,1)])
provref("<or=",[r(5,1,1),(e(4,4,1)])
pred("proud",[r(5,1,1)])
pred("bear",[r(5,1,1)])
pred("owns",[r(0,1,3),r(5,1,1)])
provref("old",[r(8,1,1)])
provref("<or=",[r(8,1,1),(e(4,4,1)])
pred("chair",[r(8,1,1)])
pred("owns",[r(0,1,1),r(8,1,1)])
yes
```

References

1. Karttunen, L.: Radical Lexicalism. Report No. CSLI 86 68. Stanford (1986).
2. Zeevat, H.: A Compositional Version of Discourse Representation Theory. Linguistics and Philosophy 12 (1991) 95 131.
3. Zeevat, H.: Aspects of Discourse Semantics and Unif. Gr. Ph.D., U. Amsterdam (1991).
4. van Eijck, J., H. Kamp: Representing discourse in context. In: van Benthem, J., A. ter Meulen (eds.): Handbook of Logic and Language. Elsevier, Amst. & MIT Press, Cambridge, Mass. (1997).
5. Chomsky, N.: The Minimalist Program. MIT Press, Cambridge, Mass. (1995).
6. Partee, B. H., G. B. ter Meulen, R. P. Wall: Mathematical Methods in Linguistics. Kluwer Academic Publ (1990).
7. Kamp, H.: A theory of truth and semantic representation. In: Groenendijk, J., T. Janssen, M. Stokhof (eds.): Formal methods in the study of language. Amsterdam, Math. Centre.
8. Groenendijk, J., M. Stokhof: Dynamic Predicate Logic. Linguistics and Philosophy 14 (1991) 39-100.
9. Chomsky, N.: Syntactic Structures. The Hague, Mouton (1957).
10. Lehmann, Ch.: On the Function of Agreement. In Barlow, M., Ch.A. Ferguson (eds): Agreement in Natural Languages. Approaches, Theories, Descriptions. CSLI Stanford (1988) 55-65.
11. Kamp, H., U. Reyle: From Discourse to Logic. Kluwer Academic Publ. (1993).
12. Alberti, G., K. Balogh, J. Kleiber, A. Viszket: Totally Lexicalist Morphology. Talk at the Sixth International Conference on the Structure of Hungarian. Dsseldorf (2002).
13. Alberti, G.: Indo-Germanic Word Order Phenomena in a Totally Lexicalist Grammar. In Sprachteorie und germanistische Linguistik 11.2. Univ. of Debrecen, Hungary (2001) 135-193.
14. Alberti, G., K. Balogh, J. Kleiber: GeLexi Project: Prolog Implementation of a Totally Lexicalist Grammar. In de Jongh, H. Zeevat, Nilsenova (eds.): Proceedings of the Third and Fourth Tbilisi Symposium on Language, Logic and Computation. ILLC, Amsterdam, and Univ. of Tbilisi.
15. Alberti, G.: Lifelong Discourse Representation Structure. In Poesio, M., D. Traum (eds.): Gothenburgh Papers in Computational Linguistics 00-5 (2000) 13-20.