Case at the Syntax/Semantics Interface

Marcus Kracht *
II. Mathematisches Institut
Arnimallee 3
D – 14195 Berlin
kracht@math.fu-berlin.de

Abstract

For most theories of semantics cases are irrelevant at best. For these theories the ideal language has fully regimented word order but no morphology. This means that cases have to disappear at the interface from syntax to semantics. The role of syntax is among other to provide a structural substitute for cases. This is also the philosophy of the Minimalist Program, where cases are treated as uninterpretable features. We argue here in line with Nordlinger [15] that case is not only independent from syntactic structure, but that it can be a substitute for it. Moreover, we shall provide a semantics in which cases are instrumental in providing the string–to–meaning translation. This shows that there is any reason to believe that cases survive at the interface.

Introduction. Semantics and to some extent also syntax are often at odds with morphology. This is so, because morphology seems to be on the one hand a small scale copy of both (we have semantic categories such as plural, and there is evidently a good deal of syntax in morphology). On the other hand, morphology

*This paper is based on a talk held at UCLA in spring 1999 and at the Symposium 'The relationship between syntax and semantics in the analysis of linguistic structure' in Helsinki 1999. I would like to thank the audiences for helpful discussions, in particular Arto Anttila, Ed Keenan, Anoop Mahajan, Ed Stabler, and Tim Stowell. Special thanks also to David Perlmutter.

often than not goes its own way. [1] These are difficult matters (more difficult than one would like to believe at first sight), and theory has often simply ignored these complications.

Morphology has received greater attention from syntacticians in the last decade, mainly because of the Minimalist Program (MP), which — in its distinct own way — links word variation and morphology, in particular case morphology. However, despite this new rapprochement between syntax and morphology, it is doubtful whether syntacticians advocating the MP are really interested in morphology. Too many questions are left aside or require an astronomic amount of derivational steps, for example agreement inside DPs. Likewise, there is at present quite limited interaction between morphology and semantical theory. However, in order to provide an integrated theory of language, we must say something about the interaction of all these three, semantics, morphology, and syntax. Recently, with referent systems, introduced by Vermeulen (see [20]), a semantics was developed that allowed — in principle — to integrate morphology into semantics. In [11], we have undertaken the effort of spelling out such a system, which links semantics not only with morphology but also with syntax. In this paper we follow the consequences of this approach with respect to the semantic role of cases. Traditionally, generative grammar (including, so it seems, the MP) assumes a split between structural cases and inherent cases. Structural cases were thought to be assigned solely by structural configurations and therefore meaningless (uninterpretable, to choose the jargon of the MP), while inherent cases can and often do carry meaning. Initially, only nominative and accusative were classified as structural, the rest being inherent. The theory never went very far in specifying the mechanics of inherent cases. In subsequent years it has emerged however that there are many more structural cases, and that there is hardly a case which is structural or inherent as such, rather, most cases can be either structural or inherent, depending on circumstances. [2] Moreover, cases can be very complex in

[1] A particularly interesting case are inflected pronouns in Hungarian. The suffix -val/-vel denotes the instrumental/comitative case. So you say a kocsival (*with the car*), where kocsi means *car*. But you do not say ⋆énnel, where én is 1st person singular pronoun. (The -v in the suffix assimilates to the adjacent root consonant, here n.) Rather, you say velem, where -em is a suffix for 1st person singular (which appears in verbal inflection, for example). Exactly the same happens with PPs involving a pronoun. Simlarly, in German you have darauf (lit. *on there-on*) and not *auf da, but auf dem Tisch (*on the table*) and not dem Tisch-auf. (The -r- infix is purely morphophonologically conditioned.) The connections between morphological selection and directionality have been a very active topic of research in syntactic theory ever since the appearance of Kayne [9], but still seem to be elusive.

[2] See Bailyn [1] for a structural instrumental in Russian and Przepiórkowski [16] and references

structure (see Mel'cuk [14]). It appears therefore that the theory of cases has to be seriously reconsidered. In this paper we shall consider the question of structural cases. In particular, we are interested in whether or not these cases are simply uninterpreted or even uninterpretable features of the syntactic processor, as claimed in MP, or whether they also carry some meaning. Our proposal is that the latter is the case. They are always meaningful, although their semantic contribution is not of model–theoretic nature. This — if true — renders the notion of a structural case redundant.

Why is there (Structural) Case? Case is a syntactic and a morphological category. [3] Theories of syntax and morphology give ample evidence of this. But it is not at all clear whether case is also a semantic category. In the Minimalist Program, for example, case is the prototypical example of a feature that serves no interpretive purpose and hence must be deleted before the structure is shipped to the syntax–to–meaning interface. Montague Grammar, similarly treats morphology simply as epiphenomenal of the constituent formation process. In MG, the string–to–meaning translation is effected in two steps:

1. Assign a binary analysis to the string.
2. Apply the semantic translation rules bottom up.

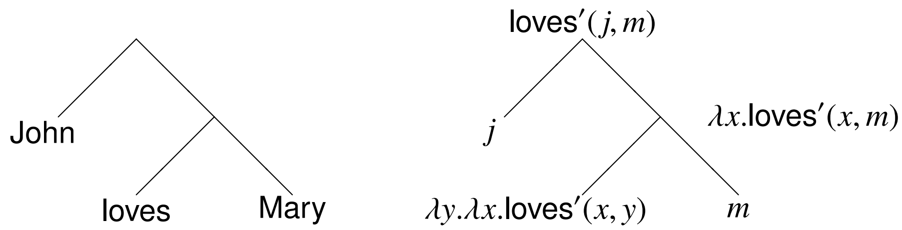
There are rules which have certain morphological side conditions, and these must be met by the analysis (see below). Take for example the string (1). It possesses the bracketings (1a) and (1b).

therein. Przepiókowski finds three structural cases in Polish: nominative, accusative amd genitive. Since all of them can also be inherent, he simply distinguishes lexical nominative, accusative and genitive from inherent nominative, accusative and genitive, respectively, in Polish. So, rather than three cases, he now has six (not to mention the inherent ones, dative, locative and instrumental). His analysis is at this point not very insightful. If all we can do is split every case in two, the problem is why there is no genuine structural case in the first place. As we shall argue below, there is reason to believe that structural cases do not exist.

[3] A note of clarification is in order. Mel'cuk [13], following traditional usage, defines case merely as a morphological category. Syntax simply uses the case distinctions for its own purpose. In our view, there is no distinction from a syntactical point of view between a case marked DP and a PP. Both can assume the role of an argument or an adjunct, depending on circumstances. This means that one should in effect consider argument PPs as if they were case marked DPs. Thus, there appear to be two kinds of cases: the ones that show up as syntactic labels, and the cases that we traditionally know of from morphology. English, for example, has no morphological dative, but one can argue that syntax does distinguish a dative (realized as to DP). See [11] on this point. So we basically claim that case, in addition to being a morphological category, is also a syntactic category. Syntax typically distinguishes more cases than morphology.

(1) John loves Mary.
(1a) [John [loves Mary]]
(1b) [[John loves] Mary]

However, since the type assignment is such that John cannot be combined with loves, the analysis (1b) is ruled out. So, John loves Mary will be translated as loves′(j, m) in virtue of there being only one analysis compatible with the syntactic types. [4]



From the standpoint of MG, there is little in addition to the semantical types that is needed in order to get both the syntax and the semantics right. A string is a grammatical sentence if and only if it has a translation if and only if it has the syntactic type t. In the original formulation, the extra ingredient was the *directionality* of syntactic types.

Montague dealt in his papers exclusively with English. Since English has only very marginal morphology, Montague did not spend much energy in integrating it into his system. [5] Cases and agreement must be treated by distinguishing many

[4]To make matters as simple as possible, we do not typeraise NPs. Here, NPs will be semantic arguments of the verb. The reader is assured that the arguments put forward here do not depend on this choice.

[5]For example, take a look at rule S14 (quoted from [3]).

> S14. If $\alpha \in P_T$ and $\varphi \in P_t$, then $F_{10,n}(\alpha, \varphi) \in P_t$, where either (i) α does not have the form $\mathbf{he}_k$, and $F_{10,n}(\alpha, \varphi)$ comes from φ by replacing the first occurrence of $\mathbf{he}_n$ or $\mathbf{him}_n$ by α and all other occurrences of $\mathbf{he}_n$ or $\mathbf{him}_n$ by $\left\{\begin{array}{c}\mathbf{he}\\ \mathbf{she}\\ \mathbf{it}\end{array}\right\}$ or $\left\{\begin{array}{c}\mathbf{he}\\ \mathbf{she}\\ \mathbf{it}\end{array}\right\}$, respectively, according as the gender of the first B_{CN} or B_T in α is $\left\{\begin{array}{l}\text{masc.}\\ \text{fem.}\\ \text{neuter}\end{array}\right\}$ or (ii) $\alpha = \mathbf{he}_k$ and $F_{10,n}(\alpha, \varphi)$ comes from φ by replacing all occurrences of $\mathbf{he}_k$ and $\mathbf{him}_k$ by $\mathbf{he}_n$ or $\mathbf{him}_k$, respectively.

different cases in the formation rules. For example, S5 is responsible for the formation of an intransitive verb from a transitive verb and an object.

> S5. If $\delta \in P_{TV}$ and $\beta \in P_T$, then $F_5(\delta,\beta) \in P_{IV}$, where $F_5(\delta,\beta) = \delta\beta$ if β does not have the form $\mathbf{he}_n$ and $F_5(\delta, \mathbf{he}_n) = \delta\ \mathbf{him}_n$.

This is rule tells us in effect that if instead of John we had used the pronoun, we would have had to use him rather than he (gender is not even discussed at this point, therefore rule S14 quoted above is even more complex). The net effect of these rules is that among other the following sentence is ungrammatical.

(2) She loves he.

This is because under the only admitted bracketing we must form a constituent [loves he], which is not an output of S5. Therefore, (2) is ruled out. Obviously, languages with rich morphology would require an enormous amount of distinct rules and/or case distinctions inside the rules. Therefore, Bierwisch [2] made the following proposal, adopted in Haider [6]. In each rule the syntactic arguments are individually anootated for morphological properties of the arguments. The argument structure for the German verb lieben (*to love*) is in this system the following.

$$\begin{array}{ccc} \lambda x. & \lambda y. & \mathsf{love}'(y,x) \\ \vdots & \vdots & \\ \text{ACC} & \text{NOM} & \end{array}$$

The intention is that whenever this structure takes an argument, then the morphological annotation associated with the relevant abstracted variable must be respected. This makes sure that the first argument with which lieben is combined will be in the accusative case. Notice that the conditions are external stipulations, there is nothing in the λ–calculus that corresponds to them and they cannot be integrated other than by typing the arguments. It is however not very helpful to think of a semantical type *nominative*, or of *accusative* even though some grammar formalisms (HPSG) invite such speculations. Morphology simply appears to be a needless complication of the sentence. The most ideal language from the standpoint of MG would be Polish or Reverse Polish Notation: all heads universally precede or follow their arguments, and the arguments come in a fixed order. This includes the word orders VSO, VOS, SOV and OSV. SVO and OSV are not uniquely readable, but the claim of MG is that the different analyses of strings do correspond to the ambiguities in meaning. So, languages with regimented word

Suffice it to say that this is still state of the art in Montague Grammar.

order appear to fit Montague Grammar rather nicely in absence of morphological complications.

Now, languages without an equivalent of case morphology have several disadvantages. The main disadvantages are the following.

- There is no possibility to drop arguments without introducing massive ambiguity. Otherwise, the sentence Love John could either mean Love John someone or Love someone John.

- No word order variation is allowed. Otherwise, both Love John Mary and Love Mary John mean the same, and are ambiguous between SVO and SOV.

- Polyvalence introduces ambiguity.

Positively speaking, if language inserts extra material in addition to the plain string, material that lets us recover the argument structure of the sentence, then there is some freedom either to drop arguments, or to vary the word order without increasing the ambiguity. This is a mathematically provable fact. This has been defended also by Vennemann [19] (see also [12] for a long discussion of this point). It is of course another question whether languages need to avoid ambiguity to the extent that it is avoided in formal semantics. But it does appear that the association of arguments to thematic roles is a particularly sensitive point.

Therefore, cases introduce redundancy into language if simply added in addition to full word order regimentation. However, as a matter of fact, cases are often *not* redundant, since languages with rich enough case morphology tend to have a relaxed word order regime. Such is the case with German.

(3) Der Vater sieht den Sohn.
the-NOM father-NOM/ACC sees the-ACC son-NOM/ACC
The father sees the son.

(4) Den Vater sieht der Sohn.
the-ACC father-NOM/ACC sees the-NOM son-NOM/ACC
The son sees the father.

(5) Die Mutter sieht die Tochter.
the-NOM/ACC mother-NOM/ACC sees the-NOM/ACC daughter-NOM/ACC
The mother sees the daughter.
**The daughter sees the mother.*

As (3) and (4) show, with the cases interchanged the subject and object change places as well. Notice that the word Vater is ambiguous between nominative, accusative and even dative (which we did not annotate here). But when it goes together with the definite determiner der the constituent must be nominative, since der is, and when it goes together with den, the constituent must be accusative since den is. German has, as opposed to English, freer word order, which in turn makes the cases non redundant. However, free word order is constrained not by the existence of abstract cases, but by their surfce realization. For example, in absence of any other information (5) is read as SOV, and not OSV. [6]

In fact, in many languages there is free word order (either free on the level of constituents (eg German) or at the level of individual words (eg Jiwarli)). In such languages, cases are absolutely essential in recovering the meaning of the sentence. For example, any permutation of the following Latin sentence is again a (grammatical) Latin sentence and with roughly the same meaning.

(6) Senātus librum cōnsulī dat.
senate-NOM book-ACC consul-DAT gives.
The senate gives the book to the consul.

Sentence (6) raises a problem for syntactic theory. How can we ensure that any permutation of it will get the same meaning? [7] Or, to be a little bit more careful:

> How can the correct association between functor and argument be ensured by syntax in presence of free word order?

The answers to this are manyfold. However, the following is — we think — an uncontroversial thesis:

> Cases serve to disambiguate sentences in presence of free word order. Lack or loss of case distinctions leads in absence of any other factors to lack or loss of free word order.

[6] It goes without saying that intonation provides extra information. This means that variants of these sentences spoken with different intonation can actually trigger OSV readings. However, in order not to be distracted from the main arguments, we shall agree to take here written sentences, not spoken ones. In that case, questions of intonation do not arise. Likewise, no emphasis in written texts is allowed either.

[7] We attempt no proof here that all these permutations do have the same meaning. We have been assured by Alan Dench (p.c.) that in some Australian languages, in particular Warlpiri and Jiwarli, no difference in meaning exists between different word orders analogous to those of (6) (on condition that they are permissible), not even a pragmatic difference. This implies that the problem exists if not for Latin then at least for *some* languages.

Now, it is compatible with this thesis that cases are there *in addition to structure* and that cases are overt indicators of the structure of a sentence which is often enough not so easily recovered from the string. So, it is conceivable that even if there are cases, there is enough structure so that the case relations can be recovered from the structure alone. This is the claim in Transformational Grammar and subsequent theories. It is our next goal to discredit this viewpoint.

The sentence (6) and its permutations will be our main concern in this paper. Up to now there exists to our knowledge few coherent syntactic theories that can treat all admissible word orders in a unified or principled way. [8]

Case is Independent of Syntactic Structure. In much of the transformational literature, structural cases are viewed as determined by the syntactic configuration. In particular, the accusative case is assigned through a structural relation between the verbal head and its complement, namely sisterhood. However, there are arguments that militate against such a view. One such argument has been put forward in Haider [6]. According to Haider, the innermost (ie most immediate) complement of the verb aussetzen is the dative complement, while for zumuten it is the accusative complement. Haider gives among other the following evidence for his claim. According to Höhle [7] the focus domain is extensible if and only if the carrier of focus is contained in the most deeply embedded phrase. Hence, focus is most neutral if the carrier is a nonscrambled complement of the verb. It is therefore predicted that the complements are in base position in (7) but not in (8) since (7) is a good answer to a question Was ist passiert? (*What happened?*) while (8) is not. Similarly with (9) and (10).

(7) Er hat die Kinder einer solchen GeFAHR ausgesetzt.
He has exposed the children to such a DANGer.

(8) Er hat einer solchen Gefahr die KINDer ausgesetzt.
He has exposed to such a danger the CHILdren.

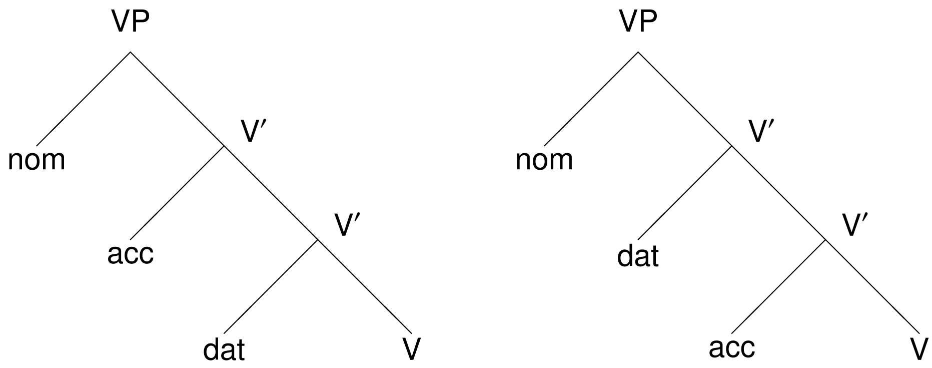
(9) Er hat seinen Studenten eine solche KlauSUR zugemutet.
He has given his students such a TEST.

(10) Er hat eine solche Klausur seinen StuDENten zugemutet.
He has given such a test to his STUdents.

It follows that some verbs select the dative complement first, others the accusative. We therefore have the following syntactic structures projected by these verbs.

---

[8] See also the discussion below. To our knowledge, LFG can cope with word order freedom of this kind.



This demonstrates that cases are not predictable from the structure alone. [9] An even stronger viewpoint is expressed by Nordlinger in [15]. She claims the following:

> *Generalization A.* Case morphology can construct grammatical relations on a par with, and independent from, phrase structure.
>
> *Generalization B.* Case morphology can construct the larger syntactic context, including and providing complex information about the clause.

For us, Generalization A is of immediate relevance. It says not only that case is independent of structure but also that case is — in some circumstances — *a substitute of syntactic structure*. In other words, there may even be languages in which there is no syntactic structure and just words. We have studied such languages from a formal point of view in [12].

Case and Word Order in Categorial Grammar. Using Categorial Grammar is one way of eliminating the syntactic defects of Montague Grammar. There are many alternative ways in categorial grammar to account for case and word order. Cases are standardly treated by annotations on the categorial labels. The verbs above receive the following type assignment:

zumuten	$(\mathrm{NP}[acc]\backslash(\mathrm{NP}[dat]\backslash(\mathrm{NP}[nom]\backslash S)))$
aussetzen	$(\mathrm{NP}[dat]\backslash(\mathrm{NP}[acc]\backslash(\mathrm{NP}[nom]\backslash S)))$

[9] In the current theory, the Minimalist Program, this argument is no longer valid, since arguments are invariably never found in their base position. This makes matters worse for the MP, since it looses the explanation for the contrast between (7) and (8) and between (9) and (10).

In Unification Categorial Grammar (UCG), category labels are complex feature structures, so that case can be specified in addition. In CCG (Combinatory Categorial Grammar, see [18]) cases are treated as type lifters. Nominative case lifts the NP to a function over intransitive verbs yielding sentence. However, if cases are independent of the number of syntactic arguments needed (as is the case in almost all languages, for example in German), this approach is bound to fail for accusative and dative. (There are verbs selecting nom–dat (gefallen, *to please*), and others selecting nom–acc (mögen, *to like*). Also the facts mentioned above militate against this view.) The problem with the first proposal is that it allows also for adjectives to change the case of the noun, an option that is not found in languages. For example, the following is the type of a premodifier changing the case of the argument from dative to accusative:

$$\mathrm{NP}[acc]/\mathrm{NP}[dat]$$

Categorial Grammar provides no principled way of excluding such things.

Fixed word order is, as is well–known, no problem for Categorial Grammar. [10] However, with word order freedom, problems start to arise. The most immediate measure that can be taken is to introduce lexical rules. However, lexical rules are too unconstrained to give us any valuable insights. It has therefore been proposed to use a nondirectional slash (here written $\multimap$), which is order insensitive. This allows for certain types of free word order, for example that of Sanskrit as proposed by Staal [17]. Staal notes that Sanskrit allows all four orders SOV, SVO, VOS and OVS. This is easily captured in order insensitive slash. However, languages with free word order of the Latin type still remain problematic. The CCG analysis of Steedman [18] needs no stipulations at all. It allows V and S to combine directly. The idea is that a subject is analysed as $V_1 \multimap S$ (here V_1, V_2, V_3 are the categories of intransitive, transitive, ditransitive verbs, respectively, an object as $V_2 \multimap V_1$. In addition to application, function composition is a rule of combination:

$$\dfrac{\begin{matrix} V \\ V_2 \end{matrix} \quad \dfrac{\begin{matrix} S \\ V_1 \multimap S \end{matrix} \quad \begin{matrix} O \\ V_2 \multimap V_1 \end{matrix}}{V_2 \multimap S}}{S}$$

[10]Certainly, for the orders SOV, SVO, OVS and VOS there is no problem. The object is the first argument and te subject the second. Only the order is specified in one or the other way. If the verb is allowed to consume its subject first — and form a constituent with it — then also VSO and OSV can be accounted for.

Still, even under these generous assumptions on typing, Latin remains a problem. The following word orders are ruled out by Steedman's system contrary to fact: IVSO and OSVI. This means that even CCG rules out some permutations of (6). [11] Having provided a negative list of how the problem is not to be solved in our opinion, we now turn to our own solution, which is based on referent systems. In order to understand why referent systems are needed, we first have to go into the detail of another semantics, namely Discourse Representation Theory.

Discourse Representation Theory. As is well known, existentially quantified variables are accessible even outside of their scope. This has led Kamp to propose a new kind of semantics, called Discourse Representation Theory (DRT) (see [8] for an introduction), in which in contrast to Montague Grammar variables appear as discourse objects and the structures are expanded simply by adding new material to the DRS. A DRS is a pair $\delta = [V : \Delta]$, where V is the *head section* consisting of a set of discourse objects and Δ is a set of *conditions*, consisting of either formulae or DRSs. 'And' is interpreted by $\bullet$, called the *Zeevat–Merge*. Formally, it is defined by

$$[V_1 : \Delta_1] \bullet [V_2 : \Delta_2] := [V_1 \cup V_2 : \Delta_1 \cup \Delta_2]$$

Here is an example.

(11) John sees a woman. She is walking.

This is translated by translating each word into a DRS and then applying the Zeevat–Merge. The last step is this:

$$\overset{\text{/John sees a woman./}}{\begin{array}{|c|}\hline x \quad y \\ \hline x \doteq j; \mathsf{woman}'(y); \\ \mathsf{see}'(x,y). \\ \hline\end{array}} \bullet \overset{\text{/She is walking./}}{\begin{array}{|c|}\hline z \\ \hline z \doteq y; \\ \mathsf{walk}'(z). \\ \hline\end{array}}$$

$$= \overset{\text{/(9)/}}{\begin{array}{|c|}\hline x \quad y \\ \hline x \doteq j; \mathsf{woman}'(y); \\ \mathsf{see}'(x,y); z \doteq y; \mathsf{walk}'(z) \\ \hline\end{array}}$$

Technically, the pronoun introduces the discourse referent z and adds the condition $z \doteq y$. The anaphoric binding potential of a pronoun is captured as follows:

[11] Steedman (p.c.) has acknowledged this point. There are apparently proposals to work with sets of arguments (as we do) in order to overcome this problem.

A DRS is *closed* if every discourse object occurring in some formula must occur in the head section of an accessible DRS.

Accessibility is a technical notion, which need not concern us here. For the moment we shall just say that it means that V contains all variables occurring in Δ. The reason why pronouns want to be bound is that all DRSs corresponding to a full text must be closed.

It has subsequently been shown that this effect can also be achieved by introducing a new semantics for predicate logic, the so–called dynamic semantics. Here, a variable can in effect be bound by an existential quantifier even outside its scope. The formal translation of (11) is then simply (12), which has the truth conditions of (13) of ordinary predicate logic:

(12) $(\exists y.\mathsf{see}'(j, y)) \wedge \mathsf{walk}'(y)$

(13) $(\exists y.\mathsf{see}'(j, y) \wedge \mathsf{walk}'(y))$

Marrying Dynamic Semantics and MG, Groenendijk and Stokhof have developed the so–called Dynamic Montague Grammar (DMG) (see [5]), which allows for dynamic binding. Dynamic Semantics has become a very dominant interpretation model in present day semantical theory.

In a series of papers, Albert Visser and Kees Vermeulen have scrutinized the methods of dynamic semantics (see among others [20] and [21]). Basically, what is lost when we step into dynamic semantics is the substitutivity of bound variables. Already, this problem has appeared in MG with pronouns. The problem is this: there is no unique logical formula or semantic unit that is substitutable for a given word in each context. Consider the following sentence.

(14) A dog sees a dog.

Suppose the DRS for a dog and see is the following.

x
$\mathsf{dog}'(x)$

$\varnothing$
$\mathsf{see}'(x, y)$

Then the translation for (14) would be the following DRS: [12]

$$\boxed{\begin{array}{c} x \\ \hline \mathsf{dog}'(x) \end{array}} \circ \left(\boxed{\begin{array}{c} \varnothing \\ \hline \mathsf{see}'(x,y) \end{array}} \circ \boxed{\begin{array}{c} x \\ \hline \mathsf{dog}'(x) \end{array}} \right)$$

$$= \boxed{\begin{array}{c} x \\ \hline \mathsf{dog}'(x); \mathsf{see}'(x,y) \end{array}}$$

However, this is contrary to fact. Instead, when the phrase a dog occurs for the second time, we must choose a different variable. Moreover, we cannot choose a variable different from y either, since we must match the variable of the object of see, which is y. For the sought for translation is this (or any alphabetic variant thereof).

$$\boxed{\begin{array}{c} x \quad y \\ \hline \mathsf{dog}'(x); \mathsf{dog}'(y); \\ \mathsf{see}'(x,y). \end{array}}$$

So, words cannot have a unique translation, rather, the variables must be carefully chosen before insertion into the translation. In fact, without a proper choice we will not get even the simplest facts right, eg the identification of subject and object. The following question therefore arises:

> Who chooses the variables for us?

DRT uses the syntactic parse as input with full indexation. So, rather than (11), what the semantic interface is given is (15):

(15) John_1 $\mathsf{sees}_{1,2}$ a_2 woman_2. $\mathsf{She}_{2,3}$ is_3 $\mathsf{walking}_3$.

Having gone this far, we are back at square one: unlike MG we do not even get the translation from the syntactic structure, we must get the indexing of the variables as well. Now, it is clear that syntax does only part of the indexing (if at all), so the whole algorithm must be supplemented by an extra component that takes care of the variable handling. In MG this component is the λ–calculus. However, MG suffers from the same problem: it does not do *all* the variable handling, only parts of it. Roughly, the set of cases handled by MG coincides with that of syntax. Now if that is so, we can either renounce the problem or start again by considering an explicit mechanism that does the handling. This has been the concern of [20] and the subsequent [21].

[12] Notice that the Zeevat–Merge is associative, commutative and idempotent. Hence, $(a \circ b) \circ a = a \circ (b \circ a) = a \circ b$.

Visser and Vermeulen argue that the problem lies with the Zeevat–Merge. By using the name x, two different DRSs are taken to be talking about the same discourse object. They argue that instead DRSs should by default be taken to talk about *different* objects. The default merge is therefore as follows.

$$[V_1 : \Delta_1] \circ [V_2 : \Delta_2] := [V_1 \times \{1\} \cup V_2 \times \{2\} : \Delta_1^1 \cup \Delta_2^2]$$

Here Δ_1^1 results from Δ_1 by replacing each variable x by $\langle x, 1\rangle$ (which we write x^1) and Δ_2^2 results from Δ_2 by replacing each variable x by $\langle x, 2\rangle$ (which we denote by x^2). This seems to make matters worse. For then (16) means something like (17):

(16) John sees Mary.
(17) *There is John. Someone sees someone. There is Mary.*

This is where the referent systems come in. Referent systems are a device for the explicit handling of variables. By choosing in addition to the usual letter x, y etc a so–called *name* for their variables, referent systems communicate with each other which variables they are using should be taken as talking about the same thing. [13] Variables with the same name will end up as the same variable in the merge, but variables with different names will end up as different variables. This is the key idea. To put this into a slogan:

> Referent Systems provide means of making the variable handling explicit rather than leaving it implicit.

This means specifically that we can use genuine linguistic information to do the variable handling and drive the translation. A *semantic structure* looks like this:

R
V
Δ

where $[V : \Delta]$ is a DRS and R a special kind of referent system, which we call the *argument structure*. This is what we shall turn to now.

Referent Systems and Argument Structure. The very basic idea is the following. A variable can be associated with a (temporary) name, which determines its

[13]In what is to follow, we will ignore the distinction between a variable and a referent. A variable is a particular member of the set of variables. However, we never write down variables, we write metavariables. This hardly creates a confusion and the distinction is hardly maintained even in logic textbooks. Likewise, we propose to be less strict in our terminology in the hope that no confusion will arise as well.

identity under merge. The set of names, denoted by N, is completely arbitrary, but for our purposes names shall be names of particular cases (*nominative*, *accusative* and so on). Let us give a temporary definition.

Definition 1 *Let V be a set of variables, N a set of names. A* ***referent system*** [14] *over N and V is a set $R \subseteq V \times N$ which is a partial injective function. This means that (a) if $\langle x, A\rangle, \langle x, B\rangle \in R$ then $A = B$, and (b) if $\langle x, A\rangle, \langle y, A\rangle \in R$ then $x = y$. We say that x* ***has name*** *A* ***in*** *R if $\langle x, A\rangle \in R$. If x has no name in R it is called* ***anonymous (in*** *R).*

It is therefore not required for a variable to have a name, a fact, which the reader is asked to keep in mind. The definition of merge is done in two stages. First, we define the merge of members of pairs $\langle x, A\rangle$, and on the basis of that we define the merge of referent systems. If two referent systems are merged, then those referents that have the same name click together. All others are treated as pairwise distinct entities (in particular referents without a name). The basic rule is this (where $B \neq A$, but possibly $x = y$):

$$\begin{array}{lcllcl} \langle x, A\rangle & \circ & \langle y, A\rangle & = & \langle x^1, A\rangle \\ \langle x, A\rangle & \circ & \langle y, B\rangle & = & \text{undefined} \end{array}$$

If referent x is anonymous, we write $\langle x, -\rangle$. The merge of two referent systems H and K is simply the set of all mergers $\alpha \circ \beta$, where $\alpha \in H$ and $\beta \in K$. The following example gives an idea of the mechanics of merge:

$$\begin{array}{|c|}\hline \langle x, nom\rangle, \langle y, dat\rangle \\ \hline x \\ \hline \varphi(x, y, z) \\ \hline \end{array} \circ \begin{array}{|c|}\hline \langle x, nom\rangle, \langle z, dat\rangle, \langle y, acc\rangle \\ \hline y \\ \hline \psi(x, y, z) \\ \hline \end{array}$$

$$= \begin{array}{|c|}\hline \langle x^1, nom\rangle, \langle y^1, dat\rangle, \langle y^2, acc\rangle \\ \hline x^1 \quad y^2 \\ \hline \varphi(x^1, y^1, z^1); \\ \psi(x^1, y^2, y^1) \\ \hline \end{array}$$

Starting from this basic idea, we now refine the definitions step by step. First, we allow referents to have sets of names, for example $\{nom, acc\}$. (For the one

[14]This is actually a much simplified referent system from the standpoint of Vermeulen [20]. However, for pedagogical purposes it is better to start with this one.

membered set $\{acc\}$ we shall write acc.) [15] This model is however overly naive. All referents in the nominative would be taken as talking about the same thing. Hence, we must assume that variables can loose their names under certain circumstances. We shall therefore annotate the names as follows: we write $\vartriangle$ if the argument structure supplies the name, $\triangledown$ if it consumes the name, and $\lozenge$ if it does both. $\lozenge$ is an abbreviation of $\{\vartriangle, \triangledown\}$. Finally, we write $\varnothing$ if it does neither. In that case the referent is as if anonymous, and by convention omitted from the argument structure. [16] We call these marks *vertical diacritics*.

$$\begin{array}{lcccl}
\langle x, \vartriangle, A\rangle & \circ & \langle y, \triangledown, A\rangle & = & \langle x, -, A\rangle \\
\langle x, \vartriangle, A\rangle & \circ & \langle y, \lozenge, A\rangle & = & \langle x, \vartriangle, A\rangle \\
\langle x, \lozenge, A\rangle & \circ & \langle y, \triangledown, A\rangle & = & \langle x, \vartriangle, A\rangle \\
\langle x, \lozenge, A\rangle & \circ & \langle y, \lozenge, A\rangle & = & \langle x, \lozenge, A\rangle
\end{array}$$

So, in each case a pair consisting of a $\vartriangle$ and a $\triangledown$ is cancelled. We also say that if the argument structure contains x with diacritic $\vartriangle$ ($\triangledown$, $\lozenge$) that it is an *x–argument* (*x–functor*, *x–adjunct*).

The next complication is that we introduce order sensitivity. If the vertical diacritic is $\triangledown$ or $\lozenge$, an additional *horizontal diacritic* is added. It is associated with the $\triangledown$ of the vertical diacritic. The horizontal diacritic is a subset of $\{\curvearrowleft, \curvearrowright\}$. We write $\circlearrowright$ instead of $\{\curvearrowleft, \curvearrowright\}$. When the $\triangledown$ and $\vartriangle$ are cancelled, the referent carrying the $\triangledown$ wants the referent carrying $\vartriangle$ to its right ($\curvearrowright$), left ($\curvearrowleft$) or on either side ($\circlearrowright$). The resulting name set is the intersection of the two original name sets. If the horizontal diacritic is $\varnothing$, no merge succeeds.

An AHS is a triple consisting of a referent, a pair of diacritics, and a set of names. It looks like this:

$$\begin{array}{lllll}
\alpha = & \langle x, & \triangledown & \curvearrowright, & acc\rangle \\
 & \text{referent} & \text{handling} & \text{directionality} & \text{name}
\end{array}$$

[15] In addition, we allow a particular DRS to carry a set of alternative referent systems. The difference between the options of allowing sets of referent systems versus allowing sets of names is picked up in passim in Footnote 19. For the introduction of sets of names already creates some awkward situations. Suppose, for example, that we have the argument structure $F = \{\langle x, \{A, B\}\rangle, \langle y, \{B, C\}\rangle\}$. Let it be merged with the structure $G = \{\langle z, B\rangle\}$. Then, we get $F \circ G = \{\langle x^1, B\rangle\}$. However, this is not what is desired. For the argument structure is a disjunction of four distinct argument structures, each of which produce different results under merge with G. However, at future stages of refinement, this problem will disappear.

[16] We have said 'as if anonymous' rather than 'anonymous'. The difference is, however, slight. The mechanics of a referent with annotation $\varnothing$ is the same as that of a referent that actually has no name. Notice that we omit anonymous as well as 'as if' anonymous referents in the argument structure, although the latter are technically speaking still present. The definitions will have to take care of this fact.

α says in informal talk: when two structures are merged, the referent x is identified with whatever referent in the other structure has the name *acc*, if that structure is to the right ($\curvearrowright$). In that case, the name is thrown away after merge ($\triangledown$). The mechanics of the vertical diacritics is a little bit like categorial grammar. Merge cancels out an occurrence of $\triangledown$ and $\vartriangle$. The variable (or the AHS) carrying $\triangledown$ is called the *merge head*, the variable (or AHS) carrying the $\vartriangle$ is called the *merge argument*. The variable of an adjunct (vertical diacritic $\Diamond$) can thus either be a merge head (when acting as a modifier), or an argument (which in configurational languages can only arise when it merges with another adjunct, see below for this case). We first do the merge of two single AHSs. Here are some illustrative examples:

$$
\begin{array}{lcccl}
\langle x, \triangledown \curvearrowright, nom\rangle & \circ & \langle y, \vartriangle \varnothing, nom\rangle & = & \langle x^1, \varnothing\varnothing, -\rangle \\
\langle x, \Diamond \curvearrowright, nom\rangle & \circ & \langle y, \vartriangle \varnothing, nom\rangle & = & \langle x^1, \vartriangle \varnothing, nom\rangle \\
\langle y, \vartriangle \varnothing, nom\rangle & \circ & \langle x, \triangledown \curvearrowleft, nom\rangle & = & \langle y^1, \varnothing\varnothing, -\rangle \\
\langle y, \Diamond \curvearrowleft, \{nom, acc\}\rangle & \circ & \langle x, \Diamond \curvearrowleft, nom\rangle & = & \langle y^1, \varnothing \curvearrowleft, nom\rangle
\end{array}
$$

The definition is unique except in the case where both vertical diacritics are $\Diamond$ and the left hand AHS has horizontal diacritic $\curvearrowright$ or $\circlearrowleft$ and the right hand AHS has horizontal diacritic $\curvearrowleft$ or $\circlearrowleft$. The problem is that we may either of the AHSs as the head.

$$
\langle x, \Diamond \curvearrowright, A\rangle \circ \langle y, \Diamond \curvearrowleft, A\rangle = \begin{cases} \langle x^1, \Diamond \curvearrowleft, A\rangle \\ \langle x^1, \Diamond \curvearrowright, A\rangle \end{cases}
$$

Therefore, the merge will be declared undefined in these cases.

Definition 2 *An **argument handling statement (AHS) over N** is a triple $\langle x, \delta, S\rangle$, where x is a variable, δ a pair consisting of a vertical diacritic and a horizontal diacritic, and $S \subseteq N$ a set of names. An **argument structure (over N)** is a sequence of AHSs over N such that (1) no variable appears twice, (2) exactly one variable has the diacritic $\Diamond$ or $\vartriangle$. This variable is called the **head** of the argument structure and its AHS is the first in the sequence. A referent is **anonymous** in an argument structure if either (a) its name is $\varnothing$, (b) its vertical diacritic is $\varnothing$ or (c) its vertical diacritic is $\triangledown$ or $\Diamond$ but the horizontal diacritic is $\varnothing$.*

We shall briefly return to the problem of merge with sets of names. Under the present definition, if α and β can be merged, then only the head variables of α and β are suitable candidates for sharing (= identification), all others must remain distinct. The merge of argument structures is described below.

Basic Syntax. An argument structure contains only one head. The head decides

Table 1: The Basic Syntactic Types

	$\vartriangle$	$\Diamond$
object	noun	adjective
event	verb	adverb

the syntactic category of the item. We have two basic semantic types (there may be more): objects and events. Now let x be the head of R. Then we assign a category to R in the following way. We have two basic dichotomies. The vertical diacritic of x can be $\vartriangle$ or $\Diamond$, and the type of x can be that of an object or that of an event. This gives four possibilities, which are summarized in Table 1. So, the basic categories fall out from the argument structure. [17] Further, there is no need to distinguish syntactically between heads and phrases. We shall assume that an argument structure is *phrasal* if it contains at most one non–anonymous variable.

For English nouns we get for example:

/dog/
$\langle x, \vartriangle \varnothing, nom\rangle$
$\varnothing$
$\mathsf{dog}'(x)$

/teacher/
$\langle x, \vartriangle \varnothing, nom\rangle, \langle y, \triangledown \curvearrowleft, gen\rangle$
$\varnothing$
$\mathsf{teach}'(x, y)$

Here is the structure of an English transitive verb (here $\star$ is some unspecific value):

/sees/
$\langle e, \vartriangle \varnothing, \star\rangle,$ $\langle x, \triangledown \curvearrowright, nom\rangle, \langle y, \triangledown \curvearrowleft, acc\rangle.$
$\varnothing$
$\mathsf{see}'(e); \mathsf{exp}'(e) \doteq x;$ $\mathsf{thm}'(e) \doteq y$

Now how are argument structures combined? We shall assume throughout that we are dealing with configurational languages.

Definition 3 *A language is* ***configurational*** *if the merge $A \circ B$ of two argument structures is defined only if either A or B is a phrase, on condition that both are*

[17]Certainly, the type of x needs to annotated somewhere. However, we shall ignore this question of detail here. See [11] for an extensive discussion.

lexemes. [18]

Now, let A and B be two argument structures. We say that $\langle\alpha,\beta\rangle$ is a *matching pair* of A and B if $\alpha \circ \beta$ is defined. This implies among other that both α and β are non–anonymous (not even 'as if' anonymous). Now, $A \circ B$ is defined iff there exists a unique matching pair of A and B; this matching pair must be such that α accesses β and β accesses α. There are two possibilities for access. Languages will differ in whether they allow G–access or E–access.

1. α *E–accesses* β in B if β is the last non–anonymous member of B.
2. α *G–accesses* β in B if β is the last member of B having the same name as α.

The difference between the two types of access is this: if the access is E–type, then the arguments in the argument structure must be discharged in the order listed.

Now we are ready to define the merge of argument structures. The readers may take a look at the explicit example in Table 3 for the mechanics of merge. Let A and B two argument structures. We wish to define the merge $A \circ B$. Let $\langle\alpha,\beta\rangle$ be the matching pair, with x the variable of α. Let first α be the merge head of $\alpha \circ \beta$. In that case, B is a phrase. Then $A \circ B$ is defined if and only if β accesses α. The resulting argument structure is defined as follows. Let B^- the result of removing β from B. (So, $B = \beta; B^-$. All variables of B^- are anonymous.) Replace all variables y in A^- by y^1, and call the result A^*. Replace all variables y in B^- by y^2, and call the result B^*. Now $A \circ B$ is the result of replacing in $A^*; B^*$ the AHS α^1 by the AHS $\alpha \circ \beta$. If the merge head is β, the merge is defined if and only if α accesses β, and the merge is defined analogously.

Word Order Variation and Access. Word order variation and configurationality are determined through access restrictions. If the access is G–type, then the configurationality is flexible, allowing for what looks like scrambling (if arguments are uniformly on one side). Table 2 presents a nonexhaustive list of the possibilities for word order variation: See [17] for word order in Sanskrit, [10] for Berber and Toba–Batak. Word order in English and German are folklore (or

[18] This is just a sketchy version. In [11] a distinction is made between functional and nonfunctional elements. The latter are subject to the configurationality restriction. For non–configurational languages one needs to investigate the possibilities further. It is noted in [11] that simply lifting the requirement that complements must be phrases does not give enough freedom in some cases. The framework of [12] on the other hand is completely liberal, but again inadequate for the reason that non–configurational languages are not entirely without word order restrictions either. Much work is still needed to calibrate the generative capacity of these formalisms.

Table 2: Access Restrictions and Language Type

Access	Directionality		Word Order	Language
E	S: $\curvearrowleft$	O: $\curvearrowright$	SVO	English
E	O: $\curvearrowright$	S: $\curvearrowright$	VSO	Berber, Toba–Batak
E	S: $\circlearrowleft$	O: $\circlearrowleft$	{SVO, SOV, VOS, OVS}	Sanskrit
G	S: $\curvearrowleft$	O: $\curvearrowright$	{S(VO), (SV)O}	?
G	S: $\curvearrowleft$	O: $\curvearrowleft$	SOV+scrambling	German
G	S: $\circlearrowleft$	O: $\circlearrowleft$	Free	Latin, Ancient Greek

see [6]). In standard grammars of Latin and Ancient Greek one will find support for the claim that they have free word order. Otherwise, free word order has been claimed for Wambaya in [15]. (See also references therein.) We have not subsumed this language under the heading since it is probably not configurational in our sense. There are a number of possibilities that have not been tried. We may — for example — argue that English allows both S(VO) and (SV)O, whereby quantifier raising from direct object over the subject is made possible. This can be implemented in the following way. We may allow to define access in such a way that it is either directionality insensitive or directionality sensitive. The previous definitions of access can be classified as directionally insensitive. The directionally sensitive versions are as follows:

1. α *ED–accesses* β in B if β is the last non–anonymous member of B with the same horizontal diacritic as α.

2. α *GD–accesses* β if β in B is the last non–anonymous member of B having the same name as α and the same horizontal diacritic.

It is easy to see that in a verb peripheral language (a language with word order SOV, OSV, VSO, VOS or combinations thereof) there is no difference between E–access and ED–access and likewise no difference between G–access and GD–access for the verbal arguments. Only in verb medial constructions it is manifest, as in English. It is therefore predicted that German will not allow for object quantifier raising, if the latter is a corollary of the different syntactic structure. Indeed, German does not seem to allow for object quantifier raising. We are now left with a more fine grained notion of access. It is not clear yet whether this distinction has any significance beyond the facts just noted. This needs close investigation.

So, not only basic word order is accounted for but also word order freedom. Take the following permutation of (6), which is problematic for CCG:

(18) Cōnsulī dat senātus librum.

Its semantic interpretation is shown in Table 3. We have chosen here the bracketing $(A_1 \circ (A_2 \circ A_3)) \circ A_4$. A different bracketing would have resulted in an alphabetic variant of the structure for (18). The second line of Table 2 is also interesting. If a language has VSO order, then the only way to assign a constituent structure is (VS)O. So the verb must have the subject lower in the argument structure. This is in line with Keenan [10].

One consequence is worth noting. The availability of free word order depends in this model on the existence of overtly distinct cases. From a speakers point of view, the labels are distinct, so (5) should be parseable either as SOV or as OSV. However, from a hearers point of view, only the reading SOV will be generated. For if a complement could be the object, then by the rules of G–access it is. There is no choice. Dutch has less distinctive morphology, and scrambling in Dutch is more restricted than in German. [19]

Against Structural Case. We shall return to the question of structural case. There

[19] A delicate test case for our analysis is the following sentence.

(i) ..., dass den Vater die Mutter sieht.

This sentence is perfectly legitimate in German and receives OSV interpretation with normal intonation. Yet, our semantics predicts that it is ungrammatical. For it will make the phrase die Mutter the object, since it matches the last member of the argument structure of the verb. After merging with die Mutter it now expects a nominative argument, and there is none. (This example was brought to my attention by David Perlmutter.) The way to approach this problem is as follows. We think of a referent as having not a set of names; if it does, this is simply an abbreviation for the corresponding set of argument structures. The rest remains the same. Now the phrase die Mutter can be read both as nominative and accusative. Both options must be tried. If we take it to be object, then we fail. If we take it to be object, however, we succeed. So, the local ambiguity is resolved at a later stage. The choice is now a global choice, not a local one. This conforms well with an observation of Roman Jakobson, that case is global phenomenon, not a local one; which is to say that we must look at the entire sentence before we know what the parts mean.

If this line is taken, however, we reach the conclusion that (5) is simply ambiguous. The reason that it is not lies in the fact that German is a subject first language. The hearer will in all likelihood take the first DP to be the subject unless this fails. This will then explain why (i) is better than (ii).

(ii) ..., dass die Mutter der Vater sieht.

Although the phrase die Mutter can be both subject and object, the hearer will first try the hypothesis that it is subject. It is this latter assumption on hearer strategies that makes all the difference here. Without it, all word orders ould be equally plausible in German, and surface distinctness of cases would play no role whatsoever. Notice that such languages do exist!

Table 3: Interpreting IVSO Word Order

$$
\left(
\begin{array}{c}
\text{/cōnsulī/} \\
\begin{array}{|c|}\hline \langle x, \vartriangle \emptyset, dat\rangle \\ \hline \end{array} \\
\begin{array}{|c|}\hline x \\ \hline \text{consul}'(x) \\ \hline \end{array}
\end{array}
\circ
\left(
\begin{array}{c}
\text{/dat/} \\
\begin{array}{|c|}\hline \langle e, \vartriangle \emptyset, \star\rangle, \langle x, \triangledown \circlearrowright, nom\rangle, \\ \langle y, \triangledown \circlearrowright, dat\rangle, \langle z, \triangledown \circlearrowright, acc\rangle. \\ \hline \end{array} \\
\begin{array}{|c|}\hline \emptyset \\ \hline \text{give}'(e); \text{thm}'(e) \doteq y; \\ \text{agt}'(e) \doteq x; \text{goal}'(e) \doteq z. \\ \hline \end{array}
\end{array}
\circ
\begin{array}{c}
\text{/senātus/} \\
\begin{array}{|c|}\hline \langle x, \vartriangle \emptyset, nom\rangle \\ \hline \end{array} \\
\begin{array}{|c|}\hline x \\ \hline \text{senate}'(x) \\ \hline \end{array}
\end{array}
\right)\right)
$$

$$
\circ
\begin{array}{c}
\text{/librum/} \\
\begin{array}{|c|}\hline \langle x, \vartriangle \emptyset, acc\rangle \\ \hline \end{array} \\
\begin{array}{|c|}\hline x \\ \hline \text{book}'(x) \\ \hline \end{array}
\end{array}
$$

$$
=
\left(
\begin{array}{c}
\text{/cōnsulī/} \\
\begin{array}{|c|}\hline \langle x, \vartriangle \emptyset, dat\rangle \\ \hline \end{array} \\
\begin{array}{|c|}\hline x \\ \hline \text{consul}'(x) \\ \hline \end{array}
\end{array}
\circ
\begin{array}{c}
\text{/dat senātus/} \\
\begin{array}{|c|}\hline \langle e^1, \vartriangle \emptyset, \star\rangle, \langle y^1, \triangledown \circlearrowright, dat\rangle, \langle z^1, \triangledown \circlearrowright, acc\rangle. \\ \hline \end{array} \\
\begin{array}{|c|}\hline x^1 \\ \hline \text{give}'(e^1); \text{thm}'(e^1) \doteq y^1; \text{agt}'(e^1) \doteq x^1; \\ \text{goal}'(e^1) \doteq z^1; \text{senate}'(x^1). \\ \hline \end{array}
\end{array}
\right)
$$

$$
\circ
\begin{array}{c}
\text{/librum/} \\
\begin{array}{|c|}\hline \langle x, \vartriangle \emptyset, acc\rangle \\ \hline \end{array} \\
\begin{array}{|c|}\hline x \\ \hline \text{book}'(x) \\ \hline \end{array}
\end{array}
$$

$$
=
\begin{array}{c}
\text{/cōnsulī dat senātus/} \\
\begin{array}{|c|}\hline \langle e^{12}, \vartriangle \emptyset, \star\rangle, \langle y^{12}, \triangledown \circlearrowright, dat\rangle. \\ \hline \end{array} \\
\begin{array}{|c|}\hline x^{12}, \quad x^{1} \\ \hline \text{give}'(e^{12}); \text{thm}'(e^{12}) \doteq y^{12}; \text{agt}'(e^{12}) \doteq x^{12}; \\ \text{goal}'(e^{12}) \doteq x^1; \text{senate}'(x^{12}); \text{consul}'(x^1). \\ \hline \end{array}
\end{array}
\circ
\begin{array}{c}
\text{/librum/} \\
\begin{array}{|c|}\hline \langle x, \vartriangle \emptyset, acc\rangle \\ \hline \end{array} \\
\begin{array}{|c|}\hline x \\ \hline \text{book}'(x) \\ \hline \end{array}
\end{array}
$$

$$
=
\begin{array}{c}
\text{/(16)/} \\
\begin{array}{|c|}\hline \langle e^{121}, \vartriangle \emptyset, \star\rangle. \\ \hline \end{array} \\
\begin{array}{|c|}\hline x^{121} \quad y^{121} \quad x^{11} \\ \hline \text{give}'(e^{121}); \text{thm}'(e^{121}) \doteq y^{121}; \text{agt}'(e^{121}) \doteq x^{121}; \text{goal}'(e^{121}) \doteq x^{11}; \\ \text{consul}'(x^{11}); \text{book}'(y^{121}); \text{senate}'(x^{121}). \\ \hline \end{array}
\end{array}
$$

are two genuine properties of structural case. It does not contribute to semantics, and it is merely dictated by the syntactic configuration. In particular, it is not assigned by a lexical item. A structural case, by contrast, has meaning and/or is idiosyncratically assigned. As concerns the semantical contribution of structural case, we have just seen that there is no need to assume that structural case is semantically vacuous. All that needs to be acknowledged is that semantical representations contain more than just the model–theoretic information. In our case, it contains identification names for referents. These are needed to handle intersentential anaphors, to give just one example. Anyone who claims that semantical representations should not contain more than the model–theoretical meaning is asked to consider the kinds of cues a text gives to ensure the correct resolution of an anaphor. It spans from semantic information (definite dscriptions), morphological information (gender and number in pronouns) to purely textual information (the contrast between the former and the latter is an example). If semantics contains purely the model–theoretic meaning, there must be some other mechanism which takes care of the resolution of anaphors. [20] However, if the ambitions of Dynamic Montague Grammar and other semantic frameworks point into the right direction, there is no need to assume this. We conclude from this discussion that it is not problematic, it is even desirable if semantical representations contain more than the 'classical' model–theoretical meaning. This in turn means, as we have shown above, that structural cases are as interpretable as any other inherent cases.

The second characteristic property of structural cases are that they are assigned by the structure. For example, the direct complement of a verb is assigned accusative, irrespective of any other properties that it might have. We have explained earlier that this view does not seem to be correct. But suppose it were. We must first of all acknowledge that it is not the accusative alone that is structural. The specifier of a noun phrase is in the genitive in many languages, and so is the direct object of a nominalized verb. It appears that there exist many syntactic configurations which call for a (construction specific) structural case. It has been argued by Stephen Wechsler in [22] that argument structures of verbs are not as idiosyncratic as they appear at first sight. They tend to follow certain patterns. Verbs of communication, verbs of propositional attitude, movement verbs and so tend to encode certain arguments in a uniform way. Therefore, it is legitimate to postulate a kind of abstractional scale (called subsumption hierarchy), which begins with

[20] It can be argued that even the typical pragmatic categories such as speaker, hearer, indices in general, and the even the text as is must be part of the 'model–theoretic meaning', otherwise no model–theoretic account for I, you, or the expression the former. Although this blurs the distinction between semantics and pragmatics, it seems unavoidable.

the concrete verbs to run, and continues with the more abstract verb to move, to the even more abstract to do. The more concrete verbs inherit the case frames from the more abstract ones. For example, to run inherits all properties of the verb to move, which in turn inherits all properties of the vern to do. There is obviously no reason why we should stop at this level. We can simply extend the hierarchy to all lexical entries. The structural accusative is accounted for as follows. We posit an abstract transitive verb T, which has no (or next to no) specific meaning but whose argument structure contains an accusative marked complement. A verb is transitive iff it is subsumed under T in the subsumption hierarchy. Structural case is under this view only the extreme case of the argument frame inheritance principle just sketched. By being a transitive verb, the verb must assign accusative case to its complement. By being a noun, it assigns genitive to its specifier, and so on.

Case Morphology and Case Agreement. The model presented so far would perhaps be of limited interest if it could only handle the association between the head and its arguments. However, as it turns out, the model has far greater potential. By way of illustration, we shall show how DP internal agreement is accounted for. Languages typically mark either the entire DP once for case (group marking), or require other elements inside it to carry case–agreement suffixes (word marking). The Hungarian, Japanese and Turkish are group marking, Finnish, German, and Latin are word marking in this sense. Typically, group marking languages mark the nominal head, which is often peripheral in the phrase. The languages just quoted all have the nominal head at the right periphery of the DP. Now, nouns exist in various cases. To account for that, we shall assume that nouns and adjectives (and other relevant lexemes) do not carry any specific case in the lexicon, rather they carry null case. This null case is represented by the symbol $\star$. Hence the set of formal case names for German is

$$\{nom, gen, dat, acc, \star\}$$

Notice that to have the case name $\star$ is not the same as being anonymous, as will be seen immediately. Nouns therefore have the following structure in the lexicon:

/Hund/

$\langle x, \vartriangle \varnothing, \star \rangle$
$\varnothing$
$\mathsf{dog}'(x)$

For each morphological case there is a morpheme that changes $\star$ into the corresponding case name. Here we assume that functional elements have the power to change the name of a referent (which lexical elements cannot do).[21] This is indicated by the symbol $A \mapsto B$:[22]

$$\begin{array}{c}/\textsc{nom}/\\ \hline \multicolumn{1}{|c|}{\langle x, \Diamond \curvearrowleft, \bigstar \mapsto nom\rangle} \\ \hline \multicolumn{1}{|c|}{\varnothing} \\ \hline \multicolumn{1}{|c|}{\varnothing} \\ \hline \end{array}$$

Case is a suffix (it has the diacritic $\curvearrowleft$), and we get for example:

$$\begin{array}{c}/\mathsf{Hund}^\frown\textsc{nom}/\\ \hline \multicolumn{1}{|c|}{\langle x, \vartriangle \varnothing, nom\rangle} \\ \hline \multicolumn{1}{|c|}{\varnothing} \\ \hline \multicolumn{1}{|c|}{\mathsf{dog}'(x)} \\ \hline \end{array}$$

These case morphemes can also be added to adjectives:

$$\begin{array}{c}/\mathsf{blau}/\\ \hline \multicolumn{1}{|c|}{\langle x, \Diamond \curvearrowright, \bigstar\rangle} \\ \hline \multicolumn{1}{|c|}{\varnothing} \\ \hline \multicolumn{1}{|c|}{\mathsf{blue}'(x)} \\ \hline \end{array} \circ \begin{array}{c}/\textsc{nom}/\\ \hline \multicolumn{1}{|c|}{\langle x, \Diamond \curvearrowleft, \bigstar \mapsto nom\rangle} \\ \hline \multicolumn{1}{|c|}{\varnothing} \\ \hline \multicolumn{1}{|c|}{\varnothing} \\ \hline \end{array}$$

$$= \begin{array}{c}/\mathsf{blau}^\frown\textsc{nom}/\\ \hline \multicolumn{1}{|c|}{\langle x, \Diamond \curvearrowright, nom\rangle} \\ \hline \multicolumn{1}{|c|}{\varnothing} \\ \hline \multicolumn{1}{|c|}{\mathsf{blue}'(x)} \\ \hline \end{array}$$

So, /blau⁀NOM/ and /Hund⁀NOM/ can be merged to form an expression which is spelled out as blauer Hund. However, we could also merge blau and Hund without adding the case suffix. Although this is illegitimate in German, the corresponding example is grammatical in Hungarian.

[21]Compare the discussion between functional and nonfunctional elements in Footnote 18.

[22]This calls for an explanation. We have earlier accused Categorial Grammar of allowing these entities, which in general should not be admitted. Now we have introduced precisely these elements. Hence we have to be more careful. In our theory, only functional elements can change names, and only they can engage in functional merge. Lexical elements are therefore restricted both in their argument structure and in their possibilities to undergo merge. Notice also that our system is in any case far weaker than Categorial Grammar as concerns the type hierarchy.

(19) a kék kutyá-t
DET blue dog-ACC

(20) a kék kutyá-nak
DET blue dog-DAT

A way to implement this is as follows. Lexical merge can take place only between elements that are enclosed in word boundaries (#). The word boundary on the other hand can be attached only if certain (language specific) facts are met. In German, the word boundary can be added to the right of a noun or adjective only if it carries non–null case. In Hungarian, on the other hand, the word boundary can be added to a noun only if it has non–null case, and to an adjective only if it has null case. Another route is to distinguish between phrasal and word affixes.

Conclusion. Using referent systems, we see that cases (in particular structural cases) provide information about the identification of variables in the semantics. This is a decisive step ahead from previous models, which could use as their source of linguistic input only the structure, without agreement morphology. In these theories, cases had to be thrown away before syntax hands the representation over to semantics. In just the same way, the Minimalist Program takes cases as features of grammar that must disappear before the structure is shipped to the interpretive component.

In our semantics however cases are an integral part even of the representation. If this line is taken, it is now possible to use any kind of information that the structure provides for the syntax/semantics interface. For example, using φ–features a model of agreement as well as anaphoric binding can be developed.

A last note concerns the complexity of the semantics. Elsewhere (in [12] and [4]) we have presented a semantics for non–configurational languages and shown that its complexity is at most $O(n^{5/2} \log n)$. This means that parsing and semantic analysis are done very efficiently. The reason is that — unlike in Montague Grammar — substitutions are done explicitly by means of global string substitution. These substitutions can be done very fast on a computer. Without having gone through the details we expect similar positive features of the present semantic analysis.

References

[1] J. F. Baylin. Configurational case assignment in Russian syntax. *The Linguistic Review*, 12:312 – 360, 1995.

[2] Manfred Bierwisch. On the grammar of local prepositions. In M. Bierwisch, W. Motsch, and I. Zimmermann, editors, *Syntax, Semantics, and Lexicon*, Studia Grammatica XXIX, pages 1 – 65. Akademie Verlag, Berlin, 1988.

[3] David R. Dowty, Robert E. Wall, and Stanley Peters. *Introduction to Montague Semantics*. Number 11 in Synthese Library. Reidel, Dordrecht, 1981.

[4] Christian Ebert and Marcus Kracht. Formal syntax and semantics of case stacking languages. In *Proceedings of the EACL 2000*, 2000.

[5] Jeroen Groenendijk and Martin Stokhof. Dynamic predicate logic. *Linguistics and Philosophy*, 14:39 – 100, 1991.

[6] Hubert Haider. *Deutsche Syntax — generativ. Vorstudien zur Theorie einer projektiven Grammatik.* Gunter Narr Verlag, Tübingen, 1993.

[7] Tilman Höhle. Explikation für 'normale Betonung' und 'normale Wortstellung'. In W. Abraham, editor, *Satzglieder des Deutschen*, pages 75 – 133. Gunter Narr Verlag, Tübingen, 1982.

[8] Hans Kamp and Uwe Reyle. *From Discourse to Logic. Introduction to Modeltheoretic Semantics of Natural Language, Formal Language and Discourse Representation*. Kluwer, Dordrecht, 1993.

[9] Richard S. Kayne. *The Antisymmetry of Syntax*. Number 25 in Linguistic Inquiry Monographs. MIT Press, 1994.

[10] Edward L. Keenan. On semantics and binding theory. In John A. Hawkins, editor, *Explaining Language Universals*. 1988.

[11] Marcus Kracht. Agreement Morphology, Argument Structure and Syntax. Manuscript, 1999.

[12] Marcus Kracht. The Combinatorics of Cases. Manuscript, 1999.

[13] Igor Mel'cuk. Toward a definition of case. In R. D. Brecht and J. S. Levine, editors, *Case in Slavic*, pages 35 – 85. Slavica Publishers, Columbus, Ohio, 1986.

[14] Igor Mel'cuk. *Cours de Morphologie Générale.*, volume 2. Les Presses de l'Univerité de Montréal, 1993.

[15] Rachel Nordlinger. *Constructive Case. Evidence from Australian Languages.* CSLI Dissertations in Linguistics. CSLI, Stanford, 1998.

[16] Adam Przepiórkowski. *Case Assignment and the Complement/Adjunct Dichotomy. A Non–Configurational Constraint–Based Approach*. PhD thesis, Seminar für Sprachwissneschaft, Universität Tübingen, 1999.

[17] J. F. Staal. *Word Order in Sanskrit and Universal Grammar*. Foundations of Language, Supplementary Series No. 5. Reidel, Dordrecht, 1967.

[18] Mark Steedman. Gapping as constituent coordination. *Linguistics and Philosophy*, 13:207 – 263, 1990.

[19] Theo Vennemann. An explanation of drift. In C. N. Li, editor, *Word Order and Word Order Change*, pages 269 – 305. University of Texas Press, Austin and London, 1975.

[20] Kees F. M. Vermeulen. Merging without mystery or: Variables in dynamic semantics. *Journal of Philosophical Logic*, 24:405 – 450, 1995.

[21] Albert Visser and Kees F. M. Vermeulen. Dynamic bracketing and discourse representation. *Notre Dame Journal of Formal Logic*, 37:321 – 365, 1996.

[22] Stephen Wechsler. *The Semantic Basis of Argument Structure*. Dissertations in Linguistics. CSLI Publications, Stanford, 1995.