

# Compositionality: The Very Idea

MARCUS KRACHT

ABSTRACT. Compositionality is one of the most commonly used words in linguistics. Few people have a good sense of what it actually means. Many linguistic theories actually fall short of implementing it. In this paper I shall voice my discontentment with the mainstream development and propose some solutions.

## 1. INTRODUCTION

Recently, compositionality has been enjoyed something of a renaissance in theoretical linguistics, especially through [7].

Compositionality is a very popular idea in linguistics. And it is often used to claim that one's own analysis of a phenomenon is superior everyone else's. Yet, I suspect that very few of those that use the term have any clear idea of what compositionality actually buys them, if anything. For example, in the literature there are proofs to the effect that compositionality is empirically vacuous, see [9] and [27] and is therefore more a matter of presentation (but see [26] for criticism). If any of them is right, any grammar can be massaged into compositional form and so we need not worry about it as linguists. Yet, I have argued in [14] that this thought is unfounded. The reasoning is of purely exegetical character: the way the principle is phrased cannot mean what many formalisations effectively take it to mean. Additionally, I came to the conclusion that the principle effectively assumes that we know what expressions and meanings are. Although many people would like to view questions of the latter sort to be open to arbitrage (which will give them the benefit to push their own ideas) I will suggest below that there are clear intuitions as to what is right and what is not. This continues efforts

---

Through nobody's fault, this paper has been developed largely by talking to myself. I am grateful to the workshop on compositionality in Düsseldorf for allowing me to speak (and to the organiser, Markus Wernicke, for suggesting that I should speak in Paris last June about the subject). The only problem was that I was not able to take time off so I was basically without feedback. Nevertheless, the matter took a drastic turn when I finally did give a talk at the 9th Southern California Philosophy Meeting about it (and thanks to Kai Wehmeier for insisting that I should come). For then I actually did arrive at what I consider to be a formalisation of truth conditions (if you think this is trivial, read below!). The responsibility for errors is entirely my own.

begun in [15] where I have put down constraints on syntactic operations, and the subsequent [16] where I have done the same for semantics. In what is to follow, I shall expose the basic philosophy of the research and some of its conclusions. My own belief is that compositionality is far from trivial and by no means guaranteed. There are languages which simply cannot be compositional, although I believe that they are not the ones we speak. I am convinced that rather than a “nice to have” feature of grammars, it is a “must have”. Without it we have a hard time understanding how the meanings of parts figure in the meaning of the entire sentence.

Assuming that compositionality holds gives us deep insights into the structure of language. This is very different from research into generative grammar which, following structuralist theory, used syntactic data at the exclusion of any other. GB theory actually made up for the lack of semantics by introducing semantically motivated elements such as indices, and thematic roles, insisting, however, that one should not think of them in any way as being semantic. The climax of absurdity was reached with the introduction of I-language and the subsequent dismissal of rival theories as merely concerning themselves with facts of E-language. I still fail to see how one can effectively claim to have a superior analysis without saying what the criteria are upon which such a judgement is based. Here I will offer what I consider to be the first serious contender after the substitution method: *compositionality*. To make this a convincing case, however, I have to show that it is a nontrivial principle and it actually tells us a lot about how language is structured.

Of course, one may fail to believe that language is compositional. This is not to assume that language is irrational; I can think of many noncompositional grammars that are perfectly sensible. I cannot remove all doubts about my own project. Yet, anyone who is not convinced about it will have to refrain from ever again using the term to discriminate good from bad proposals unless s/he shows me how. That much I can ask for.

## 2. EXEGETICAL REMARKS

Compositionality is the thesis of autonomy of semantics. Here is what I consider to be a commonly agreed definition (taken from [?]).

The meaning of a compound expression is a function of the meanings of its parts and of the syntactic rule by which they are combined.

Let me immediately do a little exegesis of this definition. The definition attributes meanings to expressions, and not, as is commonly done in theoretical work, to analysis terms. Although the latter is a cleaner solution, it is not hard to shift to an approach that attributes meanings to expressions

instead. Thus, for all intents and purposes, expressions can be considered the carriers of meaning. The definition also speaks of “the meanings of its parts”. The use of the definite determiner is somewhat unfortunate; it suggests that the meaning of an expression is unique (it is in the analysis term).<sup>1</sup> Again it is easy to adjust that and treat the set of all of its meanings as the object to manipulate. However, it also says that the parts already have meanings. The way I read this is as follows: a **language** is a set of signs; sign consists of (at least) an expression and a meaning. A **grammar** is a finite set of constructions to form signs. Spoken languages have computable grammars, though this requirement typically is not very strong. It is grammars that are compositional or not. A language is compositional if it has a compositional grammar. This means that we can name a finite set of effective operations (called **modes**) which generate the language in question. Thus, the meanings are given beforehand, they are not subject to arbitrary choice on our part. Contrast this with [9], Page 427: “The choice of what the meaning of a part is, might depend on what we consider a suitable ingredient for building the meaning of the whole expression.” This does not go as far as saying that we can do anything we please but it also does not reject that we may adjust the meanings of parts as long as we get the right sentential meanings. This will become an important issue later on. The last word to be subjected to scrutiny is that of a “part”. [9] simply says the grammar determines what the parts are. Although this is technically correct, what I want to suggest is that it cannot proclaim anything it pleases to be parts of an expression. For a long time linguists have experimented with operations that allow deletion of material. The problem with deletion is that it obscures the part-of-relation. If anything can be done to strings, then you may claim that *compassion* is part of *wrench*. There is nothing that rules that out a priori.

### 3. GIVING IT PREDICTIVE POWER

It is a popular way to claim superiority of one’s own analysis by claiming that it is compositional. But what does it mean that an approach is compositional? For example, we shall see below that the semantics for TAGs proposed by [11] is actually not compositional despite their own claims. But what is the rationale of saying that it isn’t? The problem is, I claim, a misconception of what semantics and syntax actually are. If anything,

---

<sup>1</sup>From a linguistic viewpoint, the plural is ambiguous between saying that we deal with expressions each of which has a single meaning (and so several meanings exist solely because we have several expressions) and saying that the function takes the totality of the meanings. The latter is evidently not intended since it would allow for an expression ambiguous between meanings *A* and *B* to receive a radically different treatment than expressions that have meaning *A*, or *B*, respectively.

semantics should deal with meaning and syntax (together with morphology and phonology) with form. If one is allowed to spy into what the other is doing, compositionality becomes vacuous.<sup>2</sup> Let's look into the problem. Suppose that  $L$  is a language, that is, a set of pairs  $\langle \vec{x}, \mu \rangle$ , where  $\vec{x}$  is, say, a string, and  $\mu$  its meaning. For simplicity we assume that  $L$  is unambiguous, that is,  $\langle \vec{x}, \mu \rangle, \langle \vec{x}, \mu' \rangle \in L$  implies  $\mu = \mu'$ . Now let's change the language somewhat and put

$$(1) \quad L' := \{ \langle \vec{x}, \langle \vec{x}, \mu \rangle \rangle : \langle \vec{x}, \mu \rangle \in L \}$$

There is a straightforward compositional grammar for this language if there is a grammar for the string language. Suppose that there is a grammar  $G$ , consisting of a finite set  $F = \{f_i : i < n\}$  of string functions that generates the language  $\pi[L] := \{\vec{x} : \text{there is } \mu : \langle \vec{x}, \mu \rangle \in L\}$ . All we need is the following. Let  $h$  be the (unique function) such that for all  $\vec{x} \in \pi[L]$ :  $\langle \vec{x}, h(\vec{x}) \rangle \in L$ . Now let  $f_i$  be an  $m$ -ary function on strings. Let  $\sigma_j = \langle \vec{x}_j, \langle \vec{x}_j, \mu_j \rangle \rangle$ . Then put

$$(2) \quad f_i^*(\sigma_0, \dots, \sigma_{m-1}) := \langle f_i(\vec{x}_0, \dots, \vec{x}_{m-1}), \langle f_i(\vec{x}_0, \dots, \vec{x}_{m-1}), h(f_i(\vec{x}_0, \dots, \vec{x}_{m-1})) \rangle \rangle$$

This compositional in the sense that on both sides, strings and “meanings”, it is independently defined. The semantic function does not need to look into what the string is in order to know what it has to do. And this is for a trivial reason: the string is already present in the “semantics”. As trivial as this seems, the semantics of [11] factually does encode structural information into the semantics.

The converse is also often observed. In generative grammar the leading idea is that syntax is autonomous (the counterpart of compositionality saying that the modes of composition in syntax pay no attention to the semantic properties); moreover, once the derivation is complete it has also compiled a logical form for the expression. This means that syntax is actually doing part of the job that semantics is supposed to do, not to mention the fact that minimalist grammars use an armada of functional categories whose labels are actually of completely semantic nature. Define

$$(3) \quad L'' := \{ \langle \langle \vec{x}, \mu \rangle, \langle \vec{x}, \mu \rangle \rangle : \langle \vec{x}, \mu \rangle \in L \}$$

This is a completely symmetrified language in which syntax contains a full record of semantics and conversely. This allows compositionality to be obtained in a completely trivial fashion: syntax has all information it needs anyway, and likewise for semantics.

---

<sup>2</sup>Some theories, HPSG is an instance, do not distinguish a semantic level from a form level. There the notion of compositionality does not make much sense and I shall therefore refrain from commenting on HPSG.

If we reject such examples as trivial we must come up with a definition of what we think syntax and semantics are contain. Otherwise we do not know on what grounds we can exclude such pathological examples. Thus, the following two questions must be answered:

- ❶ What are semantic representations and how are they manipulated?
- ❷ What are syntactic representations and how are they manipulated?

These are grand questions indeed. We shall not in fact answer them directly; what we shall do, however, is delineate the boundaries in rough terms. There are things about which we can say with certainty that they do not belong to syntax: indices are a case in point, and so are  $\theta$ -roles. Likewise, there are things about which we can say with certainty that they do not belong to semantics: any reference to order and multiplicity is beyond the scope of semantics. (Linear logic disagrees with the latter statement; but see below.)

I shall also develop some positive notion of what I do think possible semantic representations look like and how they may be manipulated. Similarly for syntax. What will emerge is that if we believe that syntax and semantics are genuinely separate and that compositionality holds then we can actually get a window into the sentence structure; for we shall show that certain meanings for sentences cannot be obtained other than by assuming a particular sentence structure. This is like the dream come true for the linguist: that we need not actually refer to hopelessly unclear notions as “property of the mind” or I-language to establish sentence structure; rather, that we can use E-language—however *with* semantics—to do the very same thing. This eliminates much of the arbitrariness in thinking about sentence structure and language in general.

The outline of the paper is as follows. I shall start with a list of negative examples; I shall say why I think that certain frameworks fall short of embodying compositionality. After that I expose my own ideas about what I think the correct solutions should look like and point to examples in the literature.

#### 4. TYPE THEORY AND MONTAGUE GRAMMAR

Montague was the first to insist on the principle of compositionality and proposing a grammar that he claimed meets the standards. I shall discuss in depth two shortcomings that beset his approach and turn to later developments.

The first problem with Montague Grammar is that it is not what [6] calls *surface compositional*. This has to do with the rules for quantification. Montague wanted to get type raising and alternative scopes without complicating the type hierarchy. He therefore resorted to the following trick, which

has become popular in generative grammar: the verb is first fed pronouns, and these pronouns are later traded in for actual quantifiers. This allows to keep the type of the quantifier unique. However, it created the need for external bookkeeping. The pronouns are called  $he_i$ , where  $i$  is a natural number; the pronoun to be replaced by the quantifier must bear the same index as the variable the quantifier binds, otherwise the result is incorrect. After putting in the quantified expression, the original pronouns are either replaced by some quantified expression or by some overt pronoun, or indeed a silent one. I object to this mechanism on two grounds: first, it makes use of deletion (or silent elements, whichever) and thus it obscures the part-of relation for structures. Second, it replaces what looks like a single rule (applying the quantified expression to a VP denotation) into a parametrised set of rules and thus effectively makes the function base infinite. Although one can formulate a single rule, as Montague has done, the parameter it uses is not explicit in the representation (or so I claim) and thus cannot figure in its definition.

The problem of pronouns has been addressed in later developments, but the problem has basically remained. As long as there is a need for an explicit accounting device people have felt the need to use indices. Yet, indices may be a mere convenience; this is what I shall suggest below. Whether or not you use the pronoun  $he_7$  or  $he_{165}$  should really not matter, all that matters is whether you choose the same number for identical variables. I shall have more to say on this issue below.

Let me turn to a second objection against Montague Grammar, and that is the use of *types*. Many linguists and logicians seem convinced that the type system is grounded in reality. However, already at its very beginning it was laden with problems. To be able to treat individuals and quantified expressions in the same way Montague argued that names actually denoted sets of properties. Thus, even though the universe had individuals in it (to have denotations for the type  $e$ ) there was no way to refer to them as such; constants would consistently refer to the sets of properties that they satisfied. This led to the idea of type raising: each object of type  $\alpha$  could alternatively be seen as a function of type  $(\alpha \rightarrow \beta) \rightarrow \beta$  for each  $\beta$ . Although technically viable and elegant it leaves us with the simple question of whether it is the latter that is its meaning or the former. Of course, one may say it is the former; the latter is only derived, for convenience, and that it is derived through a mode of composition. Notice, though, that the type raising mode is also parametric (with parameter  $\beta$ ). This creates problems for its formulations (see [15]) but they can also be overcome (see [23] for a discussion within Combinatory Categorical Grammar).

The preceding discussion has shown that type assignments are far from being unique in semantics. This in itself is not the source of complaint so

long as the types assigned to the meanings result from genuine semantic considerations. What I object to is adjusting the type assignment to the needs of syntax. There is an additional problem I need to raise here. Consider the Gaelic word *faic* ‘to see’. The syntax of Gaelic is VSO. Under standard assumptions of categorial grammar (indeed, also generative grammar), constituents must be continuous; thus the verb must form a constituent with the subject in Gaelic, while it is claimed to form a constituent with the object in English. There are three possible answers to this:

- ☞ The Gaelic word *faic* translates to  $\lambda x.\lambda y.\text{love}'(x,y)$  whereas English *to love* translates as  $\lambda y.\lambda x.\text{love}'(x,y)$ .
- ☞ The translation assigned in Gaelic is the same as in English, but the mode of composition is different: the verb first composes with the subject rather than the object.
- ☞ The constituent structure is the same in both languages; thus, Gaelic has a discontinuous constituent comprising the verb and its object.

The first approach is the least attractive one for many reasons. The most important one is that it denies that *faic* is to be translated as *to love* for the reason that the semantics of the two are different. The second puts the work into the modes of composition. It is disfavoured by many because if we assume it, categorial grammar is not uniform across languages. It is a basic assumption of categorial grammar that the array of modes is constant across languages (Montague assumed only forward and backward application, for example). The third option is the most attractive one. Recently, proposals along this line have been forward. One is the framework of abstract categorial grammars by Philippe de Groote (see [3]) and another is the work by Hoffman on Turkish (see [8]); the latter however does not address questions of semantics. A third proposal, somewhat similar to [3], has been put forward by myself in [15].

## 5. GENERATIVE GRAMMAR

Generative grammar is not directly compositional. What it claims, rather, is that the generative process yields a structure, LF, which can be interpreted compositionally. This is to say that the structure can be interpreted bottom-up, even though the bottom-up algorithm is not necessarily the algorithm that produced the structure in the first place. Much of recent generative grammar is actually very similar to Montague Grammar, so the criticism levelled against the latter apply more or less identically. Let me therefore seize the opportunity here to scrutinise in more detail the use of free variables, since this turns out to be a central issue. Sooner or later all approaches produce constituents of the following form, with  $e_5$  and  $e_{186}$  empty

elements.

$$(4) \quad e_5 \text{ sees } e_{186}$$

These constituents are standardly interpreted in first-order models, which are triples  $\langle M, I, \beta \rangle$  such that  $M$  is a set,  $I$  an interpretation of the constants, functions and relations, and  $\beta$  a valuation, that is, a function from variables to the domain. (4) is interpreted as follows.

$$(5) \quad [e_5 \text{ sees } e_{186}]^{\langle M, I, \beta \rangle} = \begin{cases} \top & \text{if } \langle \beta(x_5), \beta(x_{186}) \rangle \in I(\text{love}) \\ \perp & \text{otherwise} \end{cases}$$

Thus, the valuation can make the claim true or false, and similarly for the choice of indices. The problem is that this presupposes an unlimited array of pointing devices, not an innocent assumption. Suppose that in my derivation of *John loves Maria* I use different indices, say, I translate it as

$$(6) \quad e_{1001} \text{ sees } e_{34}$$

Does my VP then have a different meaning from yours if you used (4) instead? Of course not; you and I use the same concept so the two occurrences should have the same meaning wherever they occur. In generative grammar, of course, the problem is obscured by the fact that *sees* occurs above in three different constituents, all of which sound (and look) the same. So, in defense of the theory we may simply say that the innermost occurrences actually do denote the same meaning, but the VP itself has different meaning. Granted, for most variables it also does not matter since they will be quantified away later. Some of them are not, however, and they cause concern. The thought that the choice of index could matter that much is troubling. Moreover, if we communicate with each other, indices do not get transferred (they are neither visible nor audible), and so either every message is completely closed, containing no free variables, or else it is possible to replace them by something else that does the job just as well without assuming concrete choices of names. It is this latter road that we shall be taking.

## 6. ADJUNCTION GRAMMARS

Rather than building up structure from incomplete parts, adjunction operates on complete expressions and yields complete expressions again. This proposal, which had its advocates among Zellig Harris (see for example [5]), Solomon Marcus ([19]) and Aravind Joshi ([10] and much further work), comes in two varieties. The contextual grammars by Marcus define adjunctions on strings, while tree adjunction grammars define them on trees. The latter sort of grammars has proved to be more popular. Because of the more explicit structural record it can define the necessary operations more



easily. What interests us here is whether there actually is a compositional grammar using adjunction. Consider the following example, modified from [17].

- (7) The police arrested a sailor and a barman.
- (8) The police arrested a barman and a sailor.

These sentences have the same syntactic structure and are synonymous. Now apply adjunction to the last NP:

- (9) The police arrested a sailor and a Portuguese  
barman.
- (10) The police arrested a barman and a Portuguese  
sailor.

The synonymy is gone even though we have applied the same operation. Contradiction. One may of course question the assumptions I made concerning the identity of structures or the synonymy of the original sentences. Neither is a particularly helpful strategy because it does not help in addressing the basic design fault of adjunction. The problem with it that complete expressions have complete meanings associated with them. If we disassemble them at some point we have to say at which point the semantics needs to be changed. This of course is totally impossible under a Fregean view, which associates a truth value to a sentence. A truth value has not internal structure and cannot be disassembled. To be able to do the latter, we need to assume that meanings are structured entities. Now, suppose we grant that. (I am happy to assume that meanings are structured.) Suppose that the way semantics is structured is somehow similar to the syntactic structure (let's say it is a flattened image in the spirit of f-structure in LFG). In the coordinated structure above the adjective can be adjoined to two nodes, and this makes a difference in meaning. The operation of adjunction has to be mimicked by a similar operation in semantics. This operation can also target two nodes depending on what syntax is doing it must be the one or the other. The bigger the tree gets the more adjunction sites are created and the bigger the need for close coordination between syntax and semantics. This is the opposite of compositionality since it makes semantics dependent on syntax. I have performed a close analysis in [17] and reached the conclusion that if TAGs were reformed to be compositional in the sense of the word then they would actually look more like Linear Context Free Rewrite Systems (LCFRSs). Again, this points us to the option of relaxing the constituent structure.

## 7. TOWARDS A NEW BEGINNING

Let us return to the initial questions: what are meanings, what are syntactic structures and how do they work? We shall answer first the question about the identity of meanings. The answer is: meanings are truth conditions. Nothing more, and—obviously—nothing less. It seems therefore that DRT is on the right track by assuming that the representations should be something like DRSs, which in turn are formulae of predicate logic written up in a funny notation. But, as so often, things turn out to be otherwise.

In [24], Kees Vermeulen identified a very important deficit of Dynamic Semantics. Like Discourse Representation Theory in [12], DS had to rely on an external demon to insert the correct variable names. To see an example, look at

(11)      A man walked in.    Then another man walked in.

The index attached to the variable in the second sentence must in any event be different from the one chosen in the previous sentence. On the other hand, the words *a*, *man*, *walked* in the first sentence each use the same variable. How can this be accounted for? Kamp and Reyle in [12] assume that this is achieved by having the parser generate the indexing and pass that to the semantics. Vermeulen points out, though, that the implicit assumption is that every variable is global, and that merge will assume that variables are the same in both systems if they are the same string. Instead, Vermeulen offers the opposite view: variables are local by default, or “anonymous”. Unless otherwise stated, merge will make the variable sets disjoint. To present this, referent systems have the option of connecting a variable to a **name**. The name is communicated to the outside world and is visible in particular to the merge algorithm. If the referent systems have referents that have the same name, the substitution will make them the same, all others however are being made different. I have made this proposal the basis for my [13]. Yet, I have come to the conclusion that it too suffers from a deficit. It is that the names are part of the semantics; however, they encode by design morphological properties. In this way they provide semantics with a window, albeit a small one, into syntax. Compositionality does not hold.

In a somewhat similar spirit, Kit Fine has accused the semantics for predicate logic to be noncompositional. His source of complaint was that for all intents and purposes the choice of a variable does not matter. There is no difference in a proof that begins *Let PQR be a triangle* from a proof that begins *Let ABC be a triangle* as long as letters are consistently exchanged. In Fine’s words, predicate logic is not *alphabetically innocent*. I am unsure about the conclusion that this make the semantics for predicate logic noncompositional, since the variables of predicate logic

are already actually numbered, so exchanging the numbers does produce a genuinely different statement. (I also think that the semantics of predicate logic by now is codified so that it is part of the language and cannot really be changed. But that has not always been the case.) However, in actual practice we do not use predicate logic like that. Rather, we quickly start to use metavariables. It is the choice of the latter that is arbitrary. Moreover, inasmuch as predicate logic is used to describe natural language meanings, the question becomes significant: does it make sense to have explicit variables? Fine answers the question differently from Vermeulen. He wishes to think of variables as acquiring meaning through the position at which they occur. In and of themselves, their meaning is identical. There is no way to tell the difference between  $x$  and  $y$ . It depends entirely on their context of use.

My own solution of the problem is different from the previous two. To see what its motivations are, consider the problem of teaching the concept to *stab*. One way is to give a verbal definition. Another is to point at a scene in a film or even a picture (!) and say: see, *he* (pointing your finger at Brutus) is stabbing *him* (now pointing at Caesar). You may also point out that it necessarily involves a knife, and so on. The order that the arguments find themselves in in your sentence help us keep them apart. However, it has no correlate in any ordering that is defined on the picture. There is no order in three dimensional space that corresponds to the linear order in a sentence. The association between a linear position in a sentence and a figure in a picture is determined by other factors, and they have to do with the meaning of the concept itself. When you learn the concept of *stabbing* you also learn what it is to qualify as a subject of stabbing and what it is to qualify as an object. What I assume is that the picture is actually very much the way we should think of concepts; concepts are represented as image schemata, and there is no linear order in the positions. If you want a concrete example to hold on to, think of cognitive grammar, as in [18]. Moreover, suppose you need to explain the concept *seppuku*. You could then (somewhat falsely) say: it is to stab oneself; or, again, you can point to a picture of Mishima and say: see, he is committing *seppuku*. Whether or not you think of this as a transitive action, the picture does not contain two copies of Mishima. The duplicity of variables we have in the notation is not reflected in the picture.

This leads to the following definition. Concepts are translated as relations modulo equivalence. The relations can be between objects of different types (which allows to have time points locations and so on), but we shall work here just with individuals. An  $n$ -ary relation is a subset of  $M^n$ , where  $M$  is the domain of the first-order structure. Let  $C \subseteq M^n$  be an  $n$ -ary relation and

$\pi : \{0, 1, \dots, n-1\} \rightarrow \{0, 1, \dots, n-1\}$  a permutation. Put

$$(12) \quad \pi[C] := \{\langle a_{\pi(0)}, a_{\pi(1)}, \dots, a_{\pi(n-1)} \rangle : \langle a_0, a_1, \dots, a_{n-1} \rangle \in C\}$$

$$(13) \quad d_i[C] := \{\langle a_0, a_1, \dots, a_n \rangle : \langle a_0, a_1, \dots, a_{n-1} \rangle \in C, a_n = a_i\}$$

$$(14) \quad E[C] := C \times M$$

Write  $C \approx C'$  if there is a concept  $C''$  such that  $C$  and  $C'$  can be transformed into  $C''$  using any combination of the above operations. We then put

$$(15) \quad [C]_{\approx} := \{C' : C \approx C'\}$$

To give an example, let  $M = \{a, b\}$ . The set  $\{\langle a, a \rangle, \langle a, b \rangle, \langle b, b \rangle\}$  expresses the same concept as  $\{\langle a, a \rangle, \langle b, a \rangle, \langle b, b \rangle\}$  (since we can exchange the first and second position). The concept  $\{\langle a, a \rangle\}$  expresses the same concept as  $\{\langle a \rangle\}$ , since the first and second are always identical.  $\{\langle a, a \rangle, \langle b, b \rangle\}$  expresses the same concept as  $\{\langle a \rangle, \langle b \rangle\}$ , which in turn is the same as  $\{\emptyset\}$  (using the familiar identity  $M^0 \times M \cong M$ ; they are not identical as sets, but considered identical here).

This has consequences worth pointing out. The relation denoted by `to the left of` is the inverse of the relation denoted by `to the right of`. Likewise, the relation denoted by `to be seen` is the inverse of the relation of `to see`. If the above is right, then the two pairs, although denoting different relations, actually denote the same concepts. They only differ in the way their arguments are *linked* to positions.

## 8. HOW ARE MEANINGS MANIPULATED?

Constituents denote concepts, which are sets of equivalent relations (not necessarily of same length). When two constituents are joined into a constituent, what happens to the concepts involved? The basic idea is this. Let  $C$  and  $D$  be constituents, with meaning  $\mathfrak{p}$  and  $\mathfrak{q}$ . The first stage is to pick an appropriate representative of  $P \in \mathfrak{p}$  and  $Q \in \mathfrak{q}$ ; we form the product  $P \times Q$  and intersect it with an appropriate identity, say

$$(16) \quad d_{jk}^n = \{\vec{a} \in M^n : a_j = a_k\}$$

We may additionally project to a smaller set using any number of the following functions. Finally, we take the concept that results from this relation.

What is important is that there are no variables; also, we have no first hand indication in which order the arguments are presented when we pick  $P$  and  $Q$ . What we have to do then is to find a way to make the definitions either independent of the choice of  $P$  or make sure than we can actually pick  $P$  uniquely. The latter is made possible through the introduction of a **linking aspect**. This is a function that associates with each concept  $\mathfrak{p}$  a finite set  $A$  such that there is  $P \in \mathfrak{p}$  such that  $A \subseteq P$  and if  $A \subseteq B \in \mathfrak{p}$  then  $P = Q$ . It is perhaps a surprising fact that linking aspects always exist.

Thus, the merge proceeds by temporarily dragging the variables out of their anonymity, performing identification of variables, and then letting the result sink back into anonymity. Thus, the only operation we are allowed to perform is the identification of variables. It is known that this is sufficient. Any language that has functions in it can be reduced to a relational language; the equivalent to function application becomes identification of variables. Namely, if  $f$  is the function, and if it is applied to  $x$ , then the reduction of  $f$  will produce a statement of the form  $y = f(u_0, \dots, u_{n-1})$  and “application” consists in adding the equation  $y = x$ . The set theoretic coding of functions actually does exactly that. (The variables  $x$  and  $y$  will actually end up being coded by just one position, since they share the values.)

Now, if moving from relations to concepts does not change the expressive power, why bother? One answer is that there actually *is* a difference. If  $P$  and  $Q$  are in the same concept, they will actually be treated alike. This produces delicate interactions. I have said above that the meaning of passives is the same as that of actives. They denote the same concepts but different relations. To get them to behave differently, we have to turn to their form.

## 9. SYNTACTIC REPRESENTATIONS

There is perhaps nothing more difficult as agreeing to minimal standards for syntactic representations. Yet, we have to try. Syntactic constituents have a **category** and an **exponent**. First, pushing aside a few concerns about segmentability, I consider exponents to be tuples of strings. If the constituent is continuous, we just have a single string, but to have pairs of strings is not uncommon. The idea of manipulating tuples has been reintroduced through Literal Movement Grammars in [4]. Linear Context Free Rewrite Systems are particular literal movement grammars that use a context free grammar where the exponents are tuples of strings. It is known that Minimalist Grammars in the sense of Stabler can be reduced to LCFRSs ([20]). Thus, even if one likes to think of syntactic representations as tree, there is no reason to dismiss tuples of strings as misguided. They serve the purpose just as well in Minimalism and—so I believe—elsewhere. For example [22] has shown that head grammars, a variant of 2-LCFRSs, can deal with crossing dependencies. [2] has used 2-LCFRSs to provide a compositional grammar of Swiss Grammar, which is so far the most elegant and simple solution I know of. This list can be prolonged. Basically, the weak equivalence of multicomponent TAGs with LCFRSs ([25]) is another indication. However, notice that the commitment to tuples of strings does *not* mean that we have to restrict ourselves to concatenation; a modicum of copying is in all likelihood needed ([21], whose basic insight remains valid in this connection despite [1]).

Second, once the nature of exponents is clarified, one may turn to categories. The categories form the actual grammar that links exponents with meanings (= concepts). A grammar rule of the form  $S \rightarrow NP VP$  is in actual fact translated into a binary mode  $f$  that operates as follows.  $f = \langle f^\varepsilon, f^k, f^\mu \rangle$  where  $f^\varepsilon$  is a binary function on tuples of strings,  $f^\mu$  is a binary function on concepts, and  $f^k$  a binary function on categories, defined only on the pair  $\langle NP, VP \rangle$  with result  $S$ . In English, for example,  $f^\varepsilon(\vec{x}, \vec{y}) = \vec{x} \square \vec{y}$ , but verb second in German will require a more complex function to be used.

An important principle is

IDENTITY OF INDISCERNIBLES. If  $T$  and  $U$  are  $n$ -tuples of strings that occur in the same environments, then they have the same category.

This principle says that categories should not make distinctions that go beyond the need of syntax. To see the effect of this principle, let us return to the distinction between active and passive. On the face of it, we might simply classify all active verbs as  $V[+act]$  and all passive verbs as  $V[-act]$ . This would allow active and passive verbs to link differently. However, consider a language in which passives are not syntactically distinct from actives; for example, suppose that passives are derived by having subject and object swap places (and case marking). Then, by the above principle, actives and passives cannot be distinguished by category. If that is the case, they are the same on all three levels, and the distinction disappears. In English, actives and passives actually are different syntactically. Passives fail to have a transitive object. This is the way syntax can distinguish them. Similarly, *buy* and *sell* are not mirror images of each other; their argument frames are actually quite distinct: you *buy* from, but you *sell* to. (A different story is the pair *left* and *right* which seem to be perfect mirror images of each other. At present I have no satisfying answer to that problem.)

## 10. AND HOW ARE THEY MANIPULATED?

First of all, tuples may be manipulated by concatenating their parts or by permuting their members. However, more complex operations are conceivable, the most important one being *copying*. There are, I think, genuine instances of copying, which include plural in Malay, yes-no-questions in Mandarin, and case stacking in Australian languages. This is not to say that one has to commit oneself to copying, but it is an available option. In [15] I have tried to give a few criteria of what constitutes a legitimate syntactic operation. First, there is no deletion and strings cannot be dissected. This means that every part of the tuple can be traced to at least one occurrence of this string as a substring of the entire constituent. In the case of copying there may be more. Second, there are no syncategorematic symbols. This

may be a little controversial; I know of very few exceptions to this rule. In writing, the blank is a syncategorematic symbol. In German compounds, certain sounds are inserted that have no semantic function (Fugen-s and others). Small exceptions as these aside however, no symbol is truly syncategorematic. This is important and often underestimated: without such a principle it is impossible to assume that any symbol occurring in a sentence or any part of the sentence is actually an occurrence of a constituent. Third, the exponents do not contain any empty symbols; that is to say, empty elements really leave no trace in the representation. This may be disconcerting in view of the fact that many theories (mostly variants of generative grammar) assume a plethora of empty categories. But it is not clear that their presence is really needed other than to remind the reader that they have been used in the derivation. There is nothing wrong with empty exponents, but their presence should actually be irrelevant for the definitions. For example, two signs which only differ in that one contains an empty category somewhere, are identical. Also, there is no indexation, here you cannot tell which argument has been added, and so on. Also, empty categories may not be used when invoking the principle of identity of indiscernibles. Suppose that two constituents  $C$  and  $D$  differ in that  $C$  can occur in those contexts that differ from those  $D$  can occur in except that an empty pronoun has to be added. Then the principle requires them to have identical category. Empty elements in the context make no difference. Again, all those who deny the validity of this requirement will have to tell me how they can tell “good” from “bad” use of empty elements in distinguishing categories. I have not come across an example that would necessitate giving up this stance. Again, I should stress that I do not object to the use of empty elements in representations, as long as it is clear that they are for the eye only. In other words, I contest that trees in generative grammar are stored verbatim in the head. Any alternative representation that serves the same purpose is a serious contender for “internal representation”. Until hard evidence to the contrary comes up, I will therefore remain with tuples of strings and the above rules for their manipulation.

## 11. So WHAT DOES THIS BUY US?

The present theory does complicate life at a very early stage. Already when we want to translate an innocent sentence like *Scipio stabbed Germanicus* we have to go through a lot of trouble. First, we pull out the concept of the verb, call it  $s$ . Recall that it contains plenty of relations. However, it contains only two binary relations. Since we don't know which one to pick, we need to consult the linking aspect. Let  $Y$  be the linking aspect. Now, suppose  $Y(s) = \{\langle b, c \rangle\}$ , where  $b$  is Brutus and  $c$  is Caesar. We take

the one binary relation  $P$  for which  $\langle b, c \rangle \in P$ . This assures us that whatever is the first member of a pair is the actor who stabs the second member of the pair. Now we check whether or not  $\langle s, g \rangle \in P$ , where  $s$  is Scipio and  $g$  is Germanicus. If so, the sentence is true. Otherwise it is false. The linking aspect seems like a very roundabout way to achieve this. However, it is actually very easy to apply. Suppose I have been shown a picture of Brutus stabbing Caesar; and that in addition I have been told that he (pointing at Brutus) is stabbing him (pointing at Caesar). If I have correctly grasped the concept from the scene, I have extracted an abstract image schema which I can now invoke when I hear that Scipio stabbed Germanicus. I place Scipio into the schema where Brutus had been, and Germanicus where Caesar had been. This allows me to understand what it means that Scipio stabbed Germanicus. And it allows me to say whether this is actually true. What is important is that this the entire process works without numbers or indices, it just uses positions.

I have noted in [16] that one is actually better off thinking of the linking aspect as a dynamically created object, and that this would allow for the entire process to be finitely computable. All one needs to understand is how to extend a linking aspect to new concepts. This may sound very complex but I claim that it actually is much closer to what language is really like.

However, there are also quite tangible benefits. I shall mention one, which I elaborated on in [16], namely that the present semantics predicts that Dutch has crossing dependencies. This is far from trivial. To start, all constructions involving raising infinitives can easily be claimed to have distinct meanings so that it is quite conceivable that there is a computable map from semantics to syntax. In that case there is a computable compositional grammar that is context free. In fact, we have no control over the syntactic structure whatsoever. Contrast this with the requirement that all semantics can effectively do is identify positions in relations. Then the situation changes drastically. Suppose we form a constituent between a verb, say *teach* and a noun, say *Tullius*. Then we have two options: we can identify one of the positions in the concept  $t$  of teaching with *Tullius*. This gives us the concepts of teaching *Tullius* or of *Tullius* teaching. Or we may resist identifying any two variables, in which case we get the concept of ‘teaching someone and there is *Tullius*’. In Dutch crossing dependencies, if you treat it as a nested dependency you must refrain from identifying any NP-variables with arguments of the verb until the entire verb cluster is complete. After that is done, however, we have lost any recollection of which argument appeared at which place in the structure. And so we cannot unambiguously arrive at the correct meaning.

This shows that semantics is a rather simplistic beast. It basically needs the arguments in the correct order unless there is a way to tell them apart. As



soon as the concept has two distinct positions that can be filled by different arguments we must have semantic means of telling which argument fills what place. If we don't we are lost.

## 12. CONCLUSION

The structuralist doctrine has it that syntactic structure can be assessed mainly if not exclusively through the study of syntax alone. Semantic considerations are at best viewed as a guiding hint as to where to look for evidence. I am personally not convinced that the syntactic fine structure unearthed in the Minimalist Program can be justified on anything but internal grounds, parts of which are of highly questionable theoretical status (for example, Kayne's antisymmetry thesis). By contrast, compositionality is a simple idea and is shared at least pretheoretically by linguists of many persuasions. In this paper I made an attempt to show that if we properly separate syntax and semantics then compositionality becomes a powerful tool for investigation the structure of languages.

## REFERENCES

- [1] Rajeesh Bhatt and Aravind Joshi. Semilinearity is a Syntactic Invariant. A Reply to Kracht and Michaelis 1997. *Linguistic Inquiry*, 35:683 – 692, 2004.
- [2] Mike Calcagno. A Sign-Based Extension to the Lambek Calculus for Discontinuous Constituents. *Bulletin of the IGPL*, 3:555 – 578, 1995.
- [3] Philippe de Groote. Towards Abstract Categorical Grammars. In *Association for Computational Linguistics, 39th Annual Meeting and 10th Conference of the European Chapter*, pages 148 – 155, Toulouse, 2001.
- [4] Annius V. Groenink. *Surface without Structure. Word Order and Tractability Issues in Natural Language Analysis*. PhD thesis, University of Utrecht, 1997.
- [5] Zellig S. Harris. *Mathematical Structures of Language*. Robert E. Krieger Publishing Company, Huntington, New York, 1979.
- [6] Roland R. Hausser. *Surface Compositional Grammar*. Wilhelm Finck Verlag, München, 1984.
- [7] Wilfrid Hodges. Formal features of compositionality. *Journal of Logic, Language and Information*, 10:7 – 28, 2001.
- [8] Beryl Hoffman. *The Computational Analysis of the Syntax and Semantics of "Free" Word Order in Turkish*. PhD thesis, University of Pennsylvania, 1995.
- [9] Theo Janssen. Compositionality. In Johan van Benthem and Alice ter Meulen, editors, *Handbook of Logic and Language*, pages 417 – 473. Elsevier, Amsterdam, 1997.
- [10] Aravind Joshi, Leon S. Levy, and Masako Takahashi. Tree Adjunct Grammars. *Journal of Computer and System Sciences*, 10:136 – 163, 1975.
- [11] Laura Kallmeyer and Aravind Joshi. Factoring Predicate Argument and Scope Semantics: Underspecified Semantics with LTAG. *Research in Language and Computation*, 1:3 – 58, 2003.
- [12] Hans Kamp and Uwe Reyle. *From Discourse to Logic. Introduction to Modeltheoretic Semantics of Natural Language, Formal Language and Discourse Representation*. Kluwer, Dordrecht, 1993.

- [13] Marcus Kracht. Agreement Morphology, Argument Structure and Syntax. Manuscript, 1999.
- [14] Marcus Kracht. Strict Compositionality and Literal Movement Grammars. In Michael Moortgat, editor, *Logical Aspects of Computational Linguistics '98*, number 2014 in Springer Lecture Notes in Artificial Intelligence, pages 126 – 142, 2001.
- [15] Marcus Kracht. *The Mathematics of Language*. Mouton de Gruyter, Berlin, 2003.
- [16] Marcus Kracht. The Emergence of Syntactic Structure. Manuscript, UCLA, 2005.
- [17] Marcus Kracht. Is Adjunction Compositional? Manuscript, UCLA, 2005.
- [18] Ronald W. Langacker. *Foundations of Cognitive Grammar*, volume 1. Stanford University Press, Stanford, 1987.
- [19] Solomon Marcus. *Algebraic Linguistics; Analytical Models*. Academic Press, New York and London, 1967.
- [20] Jens Michaelis. *On Formal Properties of Minimalist Grammars*. PhD thesis, Universität Potsdam, 2001.
- [21] Jens Michaelis and Marcus Kracht. Semilinearity as a syntactic invariant. In Christian Retoré, editor, *Logical Aspects of Computational Linguistics (LACL '96)*, number 1328 in Lecture Notes in Computer Science, pages 329 – 345, Heidelberg, 1997. Springer.
- [22] Carl J. Pollard. *Generalized Phrase Structure Grammar, Head Grammars and Natural Language*. PhD thesis, Stanford University, 1984.
- [23] Mark Steedman. *The Syntactic Process*. MIT Press, Cambridge (Mass.), 2000.
- [24] Kees F. M. Vermeulen. Merging without mystery or: Variables in dynamic semantics. *Journal of Philosophical Logic*, 24:405 – 450, 1995.
- [25] K. Vijay-Shanker, David J. Weir, and Aravind Joshi. Characterizing structural descriptions produced by various grammar formalisms. In *Proceedings of the 25th Meeting of the Association for Computational Linguistics (ACL '87)*, Stanford, CA, pages 104 – 111, 1987.
- [26] Dag Westerståhl. On Mathematical Proofs of the Vacuity of Compositionality. *Linguistics and Philosophy*, 21:635 – 643, 1998.
- [27] Wlodek Zadrozny. From Compositional Semantics to Systematic Semantics. *Linguistics and Philosophy*, 17:329 – 342, 1994.

DEPARTMENT OF LINGUISTICS, UCLA, 3125 CAMPBELL HALL, PO BOX 951543, LOS ANGELES, CA 90095-1543, [kracht@humnet.ucla.edu](mailto:kracht@humnet.ucla.edu)