Compositionality in Montague Grammar

Marcus Kracht
Fakultät LiLi
Universität Bielefeld
Postfach 10 01 31
D-33501 Bielefeld
marcus.kracht@uni-bielefeld.de

November 6, 2008

Abstract

Though not the first to introduce the notion, Montague has been a key figure in promoting the principle of compositionality. Montague himself proposed both an abstract program and a grammar for a fragment of English. In his fragment he makes particular choices for the manipulation of meanings that have later been modified. This work has sparked off numerous important developments. For example, Discourse Representation Theory targets a specific weakness of Montague Grammar, namely the translation of sentences as closed expressions. In this paper I shall trace the fate of the compositionality thesis in the Montague Grammar tradition.

1 Introduction

As is well known, Montague not only wrote about the Principle of Compositionality but also produced concrete grammars to show that a compositional account of quantification (and other phenomena) is indeed possible. The papers now known as PTQ ('The Proper Treatment of Quantifiers', [Montague, 1973]), UG ('Universal Grammar', [Montague, 1970b]) and EFL ('English as a Formal Language', [Montague, 1970a]) have been eye openers for the linguistic community. They mark the birth of formal semantics. Not that there has not been any formal semantics before, and not that there have not been any similar proposals on the table, but these papers demonstrated the possibility of this approach beyond doubt. Before it has always been possible to claim that such ideas will not work. Now it was hard to deny that the idea is workable.

In the wake of the new interest in formal semantics much new research has been initiated. From an abstract viewpoint it can be classed into various categories. There is research that tries to establish the ontology of language. Over time, new kinds of entities have been introduced and studied (events, situations, plurals, measures, and so on). Then there is research that questions the particular treatment of elements; should, for example, sentences be translated into propositions, that is, closed expressions, or would it be more appropriate to allow for free variables? And, finally, there is research that takes issue with the overall framework itself. It is the latter that shall interest us here. Montague defines a translation into some logical language called IL. However, it is not possible to translate the meanings of most words like /`talk`/, /`run`/, and so on, into such a language. It is expressively weak. A simple remedy would be to add enough constants, like talk', run'. However the price to be paid is that many inferences do not come out as logical inferences. For example, it is held that the inference from /`John is a bachelor.`/ to /`John is a man.`/ is true in virtue of its meaning alone; so it should be a matter of logical form. But how can this come out if both are simply translated by some constant? There are two solutions: one is to simply define the notion of 'bachelor' in terms of 'man' (and other primitives). The other is to introduce a meaning postulate. One such example is Montague's analysis of 'seek' as 'try to find'. Other people have taken issue with the idea that Montague allows the use of deletion (Hausser). Still others wish to generalize the modes of composition. In this survey I shall pinpoint some of these developments.

2 Montague's Theory

Montague's papers have been collected in [Montague, 1974], where R. Thomason also wrote quite a readable introduction. The system is explained in great detail in [Dowty *et al.*, 1981]. In this section I shall present the version of [Montague, 1970b], henceforth UG. In it Montague proposed an abstract theory of semantics for language. Before I begin with the outline proper, I shall describe a few problems that the theory solves. The first is: what *are* meanings and by what mechanism are they to be combined? The second, clearly related question, what linguistic intuitions is the theory supposed to explain?

It is perhaps best to start with the second question. One of the most solid intuitions we have is that of logical inference. We know for example (perhaps after some reflection) that the first inference is valid and the second is not.

(1)
```
   Every man walks and talks.
   John is a man.
   ---------------------------
∴  John talks.
```

(2)
```
   Every man loves some woman.
   Mary is a woman.
   ---------------------------
∴  Some man loves Mary.
```

Now why is that so? One answer is that an inference is valid because of its logical form. For example, formalising the sentences of (1) as given in (3) makes the argument valid due to the meaning of the logical elements alone.

$$(3)\qquad \begin{array}{ll} & (\forall x)(\mathsf{man}'(x) \to \mathsf{walk}'(x) \wedge \mathsf{talk}'(x)) \\ & \mathsf{man}'(\mathsf{john}') \\ \hline \therefore & \mathsf{talk}'(\mathsf{john}') \end{array}$$

This reasoning has been used among other by Davidson (see [Davidson, 1967]). In this view the logical form is supposed to transparently show why an inference is valid. This presupposes a distinction that has frequently been made between logical words (‘$\forall$’, ‘$\wedge$’, ‘$\to$’) and nonlogical words (‘man'’, ‘talk'’). The validity of the inference (3) is independent of the particular meaning of the nonlogical words. Therefore, if (3) is the logical form of (1) the validity of the latter is accounted for by appeal to the validity of the former. And that in turn is done by appeal to standard predicate logic.

The second answer to the question is that the inference is valid simply because the words mean what they mean. For example, assume that /`man`/, /`talks`/ and /`walks`/ denote subsets of the set E of entities. Let these be M, T and W, respectively. So, M is the set of all men, T the set of all talking things, and W the set of all walking things. Then the sentence /`Every man walks and talks.`/ is true if $M \subseteq W \cap T$. Now let /`John`/ denote a single object, say j. Then /`John is a man.`/ is true if and only if $j \in M$. It now follows that $j \in W \cap T$, and so $j \in T$, which is true if and only if /`John talks.`/ is true. Let us also see why the second inference fails. Here we construct a particular situation. Let $E := \{j, c, m\}$. Assume that the meaning of /`woman`/ is $W := \{c, m\}$, and the meaning of /`man`/ is $\{j\}$. Finally, the meaning of /`loves`/ is the relation $\{\langle j, c\rangle\}$. Then every man loves some woman (namely c), but no one loves m.

One should note, that the second approach is somewhat superior to the first. For appeal to the logical form in itself is not sufficient. After all, we can raise the same question with respect to the logical form itself: why is it that the inference (3) is valid? Surely, it must be because of what the formulas mean. From a practical point of view, though, we do not need to decide between the two. Montague clearly preferred the second view (inferences are valid because of what the words effectively mean) but used a logical language (the typed λ-calculus over predicate logic) to encode the meanings.

Montague seemed to have been agnostic not only about syntax but also about semantics. Like any mathematician he did not care too much what meanings really are but only how they functioned. That he used the typed λ-calculus was mere convenience on his part; he could have chosen something else as long as the basic properties are preserved. The question however is why we should choose a logical language when we already have a natural language. The answer is that natural language sentences are ambiguous. One problematic sentences was supplied by Chomsky.

(4) `Visiting relatives can be a nuisance.`

There are two ways to understand this: the nuisance is caused by the relatives that are visiting, or by visiting the relatives. Given that the concept of visit involves two arguments: a subject and an object, we would like to be clear about who is visiting. There are many more such examples (/`clever children and parents`/, where either only the children are clever, or both children and parents). Since natural language is full of such ambiguities, one aim of the translation into a logical language is to be crystal clear about what a given sentence means and what not. In translation, the sentences are neither vague nor ambiguous. Ignoring vagueness we must ask: how is it that an ambiguous sentence is translated into an unambiguous sentence? Should the meaning of (5) be rather (6) or (7)?

(5) `Every man loves a woman.`

(6) $(\forall x)(\mathsf{man}'(x) \to (\exists y)(\mathsf{woman}'(y) \wedge \mathsf{love}'(x, y)))$

(7) $(\exists x)(\mathsf{woman}'(y) \wedge (\forall x)(\mathsf{man}'(x) \to \mathsf{love}'(x, y)))$

The answer that Montague gives is that meanings are not assigned to sentences but disambiguations. Disambiguations are abstract objects which can be spelled out in two ways: as a sentence of English and as a formula of a logical language. A sentence has as many translations as it has disambiguations.

In syntax the disambiguation is done by means of different structure. Something similar happens here. We think of the constituent /`visiting relatives`/ as formed from the elements /`visiting`/ and /`relatives`/ in two different ways. We can represent these ways abstractly by binary function symbols, say f and g, and write this: $f(\mathtt{visiting}, \mathtt{relatives})$, and $g(\mathtt{visiting}, \mathtt{relatives})$. We say f and g are *modes of composition*. This presupposes, of course, that the words in this example are basic. If not, they should in turn be composed using some modes of combination. However, even if they are basic it is not always advisable to use the words themselves as objects. For there can be homonyms (say /`bank`/) and in order to prevent lexical ambiguity we need to separate them, too. We can do this by introducing two arbitrary lexical constants, say b_0 and b_1. Later on we shall specify that they both are "spelled out" as /`bank`/. (This is done by the ambiguation relation, see below.) Finally, we observe that the constituents are of different kind, called *category*. Words of identical category can be coordinated, those of different category cannot (see [Keenan and Faltz, 1985] for an elaborate semantic theory of boolean meanings):

`man and woman`	(N & N)
`walk and talk`	(V & V)
`green and blue`	(A & A)
`in and out`	(P & P)
`if and when`	(C & C)
*`man and out`	(N & P)
*`green and if`	(A & C)

The categories will also have different kinds of meanings associated with them.

Montague therefore started with an abstract language that serves to define the syntactic objects, which then get spelled out, both in terms of sound and meaning. The abstract language already uses categories, which reflect the syntactic categories seen above, but as we shall see, indirectly also the semantics categories, or types. The key notion is that of a *disambiguated language*. This is a quintuple $\langle A, \langle F_\gamma : \gamma \in \Gamma\rangle, \langle X_\delta : \delta \in \Delta\rangle, S, \delta_0\rangle$ such that

① A is the set of proper expressions,

② Δ is a list of *categories*,

③ for every δ, X_δ is the set of basic expressions of category δ,

④ $\delta_0 \in \Delta$ is a designated category (that of declarative sentences), ⑤ For every $\gamma \in \Gamma$, F_γ is an operation on the set A,

⑥ For every $\gamma, \gamma' \in \Gamma$ and elements x_i, $i < m$, and y_j, $j < n$, $F_\gamma(x_1, \cdots, x_m) \neq F_{\gamma'}(y_1, \cdots, y_n)$, unless $\gamma = \gamma'$, $m = n$ and $x_i = y_i$ for all $1 \leq i \leq n$; also, $F_\gamma(x_1, \cdots, x_m) \notin X_\delta$ for every $\delta \in \Delta$.

⑦ S is a subset of $\{F_\gamma : \gamma \in \Gamma\} \times \Gamma^* \times \Gamma$.

We put $X := \bigcup_{\delta \in \Delta} X_\delta$. So, A is the carrier set of an algebra, the algebra $\langle A, \langle F_\gamma : \gamma \in \Gamma \rangle \rangle$. This algebra is what is known as a *free algebra*. It has the property that any map $X \to B$, where B is the carrier set of an algebra with similar signature can be extended to a homomorphism.

This definition shall be simplified as follows. A *many sorted signature* is a triple $\langle F, S, \Omega \rangle$, where F is the set of *function symbols*, S the set of *sorts* and $\Omega : F \to S^+$ a map assigning to each function symbol a sequence $\langle s_0, \cdots, s_n \rangle$. An algebra of this signature is a pair $\langle \{A_s : s \in S\}, I \rangle$ where A_s is a set for each $s \in S$, and $A_s \cap A_{s'} = \varnothing$ if $s \neq s'$; and furthermore if $\Omega(f) = \langle s_0, \cdots, s_n \rangle$ then $I(f) : A_{s_0} \times \cdots \times A_{s_{n-1}} \to A_{s_n}$. (Montague allows polymorphism, that is, he allows to assign to a function symbol a set of sequences. That can be accommodated by introducing enough new function symbols.)

The disambiguated language is thus a polymorphic many-sorted algebra. Ω is the signature. Notice that the basic expressions can be identified with 0-ary functions. It is easy to see that each member of A is the (disjoint) union of certain sets A_δ. Namely, we put $F_\gamma(x_1, \cdots, x_n) \in A_\delta$ just in case $\Omega(\gamma) = \langle \delta_0, \cdots, \delta_{n-1}, \delta \rangle$. The set A_δ additionally contains the expressions of category A_δ; thus, $X_\delta \subseteq A_\delta$.

A *language*, finally, is a pair $\langle \mathfrak{L}, R \rangle$, where $\mathfrak{L}$ is a disambiguated language and R a so-called ambiguation relation. A simple example is the following. Let Λ generate fully bracketed arithmetical expressions, like this: `(((3+5)*7)+1)`. Let $\zeta R \zeta'$ if ζ is a fully bracketed expression and ζ' results from ζ by erasing all brackets. Then `(((3+5)*7+1)` R `3+5*7+1`, but also `((3+(5*7))+1)` R `3+5*7+1`. Likewise, the expressions that Montague generates for English have plenty of brackets and variables in them which are "deleted" in the ambiguation process.

Meanings are not assigned to elements of the language but to elements of the disambiguated language, and thus ζ' has meaning m in virtue of being am ambiguation of ζ that has meaning m. The semantics is provided by an algebra $\mathfrak{B} = \langle B, \langle G_\gamma : \gamma \in \Gamma \rangle, f \rangle$, where f is a map from the basic expressions to B. This map can be uniquely extended to a map h satisfying

(8) $$h(F_\gamma(x_1, \cdots, x_n)) = G_\gamma(h(x_1), \cdots, h(x_n))$$

Here, no sorts are added. Montague calls *h Fregean* if in addition *h* is a homomorphism modulo category-to-type correspondence. This is to say that the semantic algebra is not a many sorted algebra with the same sorts, and so the notion of a homomorphism cannot be employed. Rather, each sort τ of the syntactic algebra, called *category*, is mapped to a sort $\sigma(\tau)$ of the semantic algebra, also called *type*. The sortal structure of the semantic algebra is simply an image under σ of the of the syntactic algebra. If $\Omega(F_\delta) = \langle s_0, \cdots, s_n \rangle$ then $\Omega'(G_\delta) = \langle \sigma(s_0), \cdots, \sigma(s_n) \rangle$.

Actually, Montague also developed in more concrete detail what the categories and types are. There is a set C of basic category symbols in addition to a set of category constructors ($/$ and $/\!/$). A *category* is a term constructed from C with these symbols. The basic types are likewise constructed from a basic set, which Montague gives as e (entities), t (truth values) and s (indices), using a single type constructor, which I write $\rightarrow$. The mapping σ is defined thus

(9) $\sigma(\alpha/\beta) = \sigma(\alpha /\!/ \beta) = (s \rightarrow \sigma(\beta)) \rightarrow \sigma(\alpha)$

Basic categories need not be mapped to basic types; indeed, the semantics of a common noun is that of a property of individual concepts; thus, $\sigma(\mathrm{CN}) = (s \rightarrow e) \rightarrow t$.

3 A Short History of Compositionality in Montague Grammar

It is not the aim of this paper to give a full history of Montague Grammar as such. Nevertheless, in this section I shall outline some developments so as to put the subsequent discussion into proper context.

At the time of the publication of Montague's papers the most popular version of linguistic semantics was Generative Semantics (as proposed by Jackendoff, Katz, McCawley and others). Generative Semantics did semantics essentially in a syntactic fashion: meanings where bits of representation, like CAUSE, BECOME, and RED and were combined in a tree that was subject to transformations. Thus, the sentence /`John dies.`/ would be generated as follows. First, a structure like this is generated

(10) [BECOME [NOT [ALIVE *John*]]]

Then two transformations rearrange these elements as follows

(11) [[BECOME [NOT ALIVE]] *John*]

The constituent [BECOME [NOT ALIVE]] is spelled out as /`dies`/ (modulo suitable morphological manipulations). Generative Semantics never bothered to elucidate the meanings of the upper-cased expressions in any detail; it was more concerned with lexical decompositions and capturing semantic regularities (active passive, and so on).

Montague by contrast was not concerned with syntax; he was interested in getting the meanings right. Moreover, unlike Generative Semanticists he explicated the meanings using models. The lectures and seminars by Montague have had an immediate influence. The book [Cresswell, 1973] was written after Cresswell had visited UCLA. Similarly, Partee had been taught by Montague and then continued to explore the potential of this theory in natural language semantics (see the collection [Partee, 2004]). [Dowty *et al.*, 1981] was instrumental in popularising Montague Grammar. Dowty also wrote the influential [Dowty, 1979] in which he also compares Montague Grammar with Generative Semantics, arguing that there is no incompatibility between them. It is possible to assign a model-theoretic meaning to the primitives in Generative Semantics, and it is likewise possible to perform lexical decompositions within Montague Grammar.

Soon it emerged that there is even a compositional treatment of Government and Binding Theory through a mechanism that is now known as the Cooper storage ([Cooper, 1975]). The classic source for generative grammar today, [Heim and Kratzer, 1998] clearly uses Montague's ideas. A formal semantics for the Minimalist Program in that direction has been given by [Kobele, 2006]. Today, nearly all branches of formal semantics use techniques inspired by Montague's work.

Montague's use of categorial grammar also led to a rediscovery of the work of Ajdukiewicz, Bar Hillel and Lambek, and a return of categorial syntax. The fact that categorial syntax was easily paired with a semantic analysis made it extremely attractive. This is interesting since it was Chomsky who had earlier convinced both Lambek and Bar-Hillel that phrase structure grammars were superior to categorial grammars. Categorial Grammars were explored in particular in Amsterdam, where among others Theo Janssen, and Reinhard Muskens promoted the new research agenda of *compositionality* (see [Janssen, 1983], [Muskens, 1995], based on his 1989 dissertation). It is nevertheless necessary to emphasize that the use of categorial grammar does not automatically mean that the grammar is compositional. Not all developments within Categorial Grammar directly address this issue and often the relationship with semantics is not always clearly stated. The mechanism of decomposition employed in [Steedman, 1990], for example, is incompatible. A similar problem has been noted by [Calcagno, 1995] with respect to Moortgat's analysis of quantifier scope ([Moortgat, 1993]). In both cases a string is first formed and then split into components.

Categorial Grammar couples phrase structure with category, and, via the category-to-type mapping also with meanings. This is not without problems. In standard categorial grammars there is no uniform treatment of OSV languages, since the OV-constituent cannot be formed. One answer to this problem is to relax the correspondence between hierarchical structure and linear order. This has been advocated in Abstract Categorial Grammar ([de Groote, 2001], similar proposals can be found in [Muskens, 2001] and [Kracht, 2003]). ACGs treat the phonology in the same way as the semantics: expressions are no longer just strings, they are λ-terms over the algebra of strings. There are precedents for this ([Bach and Wheeler, 1983], [Oehrle, 1988]). A different solution is to allow for discontinuous constituents, for example in the form of Linear Context Free Rewrite Systems, see [Calcagno, 1995].

4 Discussion of Montague's Framework

4.1 Some Technical Remarks

Montague did not discuss much the motivations for his proposals, except in the form of exegetical remarks and an occasional example. It is however necessary to ask what his overall system achieves and what not. We shall highlight a few points where criticism has been raised of Montague's treatment and which have led to further development. Before we can enter a detailed discussion, we shall fix a few terms of discussion. Since the formal apparatus is different from Montague's, we shall have to start again with some basic definitions. The main difference with Montague's setup is that we do not assign meanings to terms of some abstract language but generate sound meaning pairs directly.

A language is defined as set of signs. Signs are pairs $\sigma = \langle e, m\rangle$, where e is the *exponent* and m the *meaning* of σ. A *grammar* is a finite set of partial functions on signs. There is no need to have sorts; however, functions are from now on partial by default. A zeroary function is also called a *constant*. The lexicon is part of the grammar; it is the set of zeroary functions. Thus the lexicon may contain entries of the form $\langle \mathtt{run}, \mathsf{run}'\rangle$. A *mode of composition* or *mode* is a function that is not zeroary.

> Let S be a set of signs and F a set of partial functions. Then $\langle S\rangle_F$ is the least set such that if $f \in F$ is an n-ary function and σ_i, $i < n$, are in $\langle S\rangle_F$ then also $f(\sigma_0, \cdots, \sigma_{n-1}) \in \langle S\rangle_F$ (if defined).

Notice that $\langle\varnothing\rangle_F = \varnothing$ unless F contains constants. The language generated by the grammar F is simply the set $\langle\varnothing\rangle_F$.

F is *compositional* if for all $f \in F$ there is a function f^μ such that for all signs $\langle e_i, m_i\rangle$, $i < n$, if $f(\langle e_0, m_0\rangle, \cdots, \langle e_{n-1}, m_{n-1}\rangle)$ exists then there is a $\vec{y}$ such that:

$$(12) \qquad f(\langle e_0, m_0\rangle, \cdots, \langle e_{n-1}, m_{n-1}\rangle) = \langle \vec{y}, f^\mu(m_0, \cdots, m_{n-1})\rangle$$

In general, for every function f there are functions f^ε and f^μ such that

$$(13) \qquad f(\langle e_0, m_0\rangle, \cdots, \langle e_{n-1}, m_{n-1}\rangle) = \langle f^\varepsilon(\vec{e}, \vec{m}), f^\mu(\vec{e}, \vec{m})\rangle$$

Thus, a grammar is compositional if the f^μ are independent of the exponents. It is the mirror image of autonomy, which requires the f^ε to be independent of the meaning (cf. [Kracht, 2003]).

Often, signs are considered to be triples $\langle e, c, m\rangle$ with the middle part being the category. A standard formulation of such grammars assumes independence of the categories on the exponents and meanings; see [Kracht, 2003]. This is the most popular format used, but contrary to popular opinion there is not much need for the additional category (see [Kracht, 2007b] for arguments). Notice that there is no start symbol. This is no accident. Although it is possible to distinguish different kinds of expressions, the language is not simply the collection of its sentences and associated meanings. If it were that the principle of compositionality would be meaningless. There would be no way we can explain the meaning of /`A man talks.`/ in terms of more primitive elements since these are not sentences and therefore would have meaning in the language.

4.2 Arbitrary Semantics

Janssen has given a proof in [Janssen, 1997] that all recursively enumerable languages have a compositional grammar. The idea is this. Let L be the language of strings and assume that every $\vec{x} \in L$ has meaning $\mu(\vec{x})$. (The problem of ambiguity shall be put aside here.) The assumption is that the set $\{\langle\vec{x}, \mu(\vec{x})\rangle : \vec{x} \in L\}$ is recursively enumerable. Then L is r.e. and has a grammar G. We transform this into the grammar G' that generates the pairs $\langle\vec{x}, \vec{x}\rangle$. Let S' be the start symbol of G'. Finally, we add a single mode

$$(14) \qquad m(\langle\vec{x}, \vec{x}\rangle) := \begin{cases} \langle\vec{x}, \mu(\vec{x})\rangle & \text{if } \vec{x} \in L \\ \text{undefined} & \text{else} \end{cases}$$

This generates the language $\{\langle\vec{x},\vec{x}\rangle : \vec{x} \in A^*\} \cup \{\langle\vec{x},\mu(\vec{x})\rangle : \vec{x} \in L\}$, so somewhat more than was originally required. In [Zadrozny, 1994], Zadrozny eliminates all empirical content from the principle by showing that all languages (recursively enumerable or not) are compositional. His proof has generated a number of responses (among which [Westerståhl, 1998]). The biggest shortcoming of his proof is that he actually does not give a compositional account of the original language; rather, he first transforms the semantics into some other semantics, from which the original semantics is easily derived. As Westerståhl points out, Zadrozny need not have used non-well founded sets for his purpose. A variant of Janssen's idea would have sufficed. Notice that Janssen's proof is subject to the same objection: Janssen enriches the original language by the set $\{\langle\vec{x},\vec{x}\rangle : \vec{x} \in L\}$. Though the idea that language has intermediate objects is quite popular (without it there would be no transformational grammar as we know it) it is debatable whether such objects are legitimate here. For the principle of compositionality talks about the meanings of the parts, and this assumes that these meanings are givens as well. Thus, any compositional grammar of a language must pass through form meaning pairs that the language itself provides. In other words, it uses only maps from L to L without postulating further signs. It is known that even this is not much of a restriction. [Kracht, 2003] shows that if a language has numbers (as almost every natural language does) then again recursive enumerability is enough. Yet even that proof must raise suspicion. The way it works is best explained with an example. Consider the following language:

(15) $$L' = \{\langle\texttt{one}, 1\rangle, \langle\texttt{one plus one}, 2\rangle, \langle\texttt{one plus one plus one}, 3\rangle, \cdots\}$$

We write $\mathrm{nb}(\vec{y})$ for the number denoted by an expression of L'. (For example, $\mathrm{nb}(\texttt{one plus one}) = 2$.) This language is certainly compositional in the stronger sense required, and has a grammar G. Suppose English is countably infinite, that is, has the form $\{\langle\vec{x}_n, m_n\rangle : m \in \mathbb{N}\}$. Then add a unary mode m with the following action:

(16) $$m(\langle\vec{y},\mu\rangle) := \begin{cases} \langle\vec{x}(\mathrm{nb}(\vec{y})), m_\mu\rangle & \text{if defined} \\ \text{undefined} & \text{else} \end{cases}$$

There are two reasons to reject this example: first, the syntactic operation is destructive, and second, there is no obvious way in which the meaning of the input figures in the meaning of the output. Unfortunately, for meanings this is harder to diagnose.

4.3 Type Raising and Flexibility

One problem area of Montague Grammar is the idea of type raising. In order to allow names to be coordinated with ordinary NPs Montague assumed that the semantics of names is identical to that of NPs. Thus, /`John`/ no longer denotes the individual John but rather the set of properties true of John. In a standard model (where we allow to quantify over all subsets) there is a biunique correspondence between these two. [Partee, 1986] takes a somewhat different turn. The idea there is that we allow grammars to raise (or lower) a type on need. In general, if A is an expression of category α (and meaning m) type raising allows to assign it the category $\beta/(\alpha\backslash\beta)$ (and meaning $\lambda n.n(m)$) or the category $(\beta/\alpha)\backslash\beta$. (I use the notation $\gamma\backslash\delta$ for the constituents that look for a γ to their left to form a δ.) The rationale is this. Suppose that B is a constituent of category $\alpha\backslash\beta$. Then $[A\,B]$ is well-formed and a constituent of category β. Given the category of B, A can be either α or $\alpha/(\beta\backslash\alpha)$. If we choose the latter, and the meaning of the entire constituent is $n(m)$, then we must choose $\lambda m.n(m)$ for A.

This proposal has been widely adopted. Also, Categorical Grammar has adopted a similar strategy to overcome the inflexibility of categories. Rather than multiplying the base categories of words, we allow to change the category of a word on need. This was the proposal made in [Geach, 1972]. The Lambek Calculus can be seen as the end result of this procedure. In the Lambek Calculus, any continuous subpart of a constituent can be a constituent again. Using standard techniques one can associate a canonical semantics with these new constituents. The Geach Rule, for example, is nothing but function composition.

Namely, let A, B and C be constituents of category α/β, β/γ and γ, respectively. Then according to standard categorial grammar the constituents can be put together only like this: $[A\,[B\,C]]$ (assuming, of course, that $\beta \neq \gamma$). However, there are circumstances where we would want the structure $[[A\,B]\,C]$, though with identical meaning. If A, B and C have meaning m, n and o, then $[A\,[B\,C]]$ has meaning $m(n(o))$, and this should also then be the meaning of $[[A\,B]\,C]$. Geach proposes a syntactic rule to combine α/β and β/γ into α/γ. Its semantic correlate is $\circ$. For $(m \circ n)(o) = m(n(o))$, whence $m \circ n = \lambda x.m(n(x))$. In natural language, the need for this rule arises rather frequently. Certain adjectives, say /`Greek`/ or /`bald`/ are properties of individuals, but can be applied also to relational nouns. The intended meaning of /`Greek neighbour`/ is "person, who is Greek and is a neighbour".

In the Lambek Calculus, every substring of a constituent can not only be given a category but also a meaning. An exposition can be found in [Morrill, 1994].

Montague's strategy of dealing with type mismatch has been dubbed 'raising to the worst case'. In anticipation of the combinatorics of an expression we adapt its type beforehand. For it seems clear that Montague intends the name /`John`/ to denote John and not a set of properties. But the proposed semantics betrays that initial idea. Contrary to popular belief this is not an innocent move. For what Montague in fact does is to supply a compositional account of a language different from the one originally proposed (though he is mathematician enough not to introduce the initial semantics in the first place). In the Categorial Grammar it is often suggested that type raising is for free. Seen from the angle of compositionality, it is not. Moreover, while Frege argues that the denotation of a verb is something of a function that takes an individual as input it is not easy to swallow the idea that the denotation of noun can equally well be seen as a function taking verb denotations. This means, namely, that the functor argument articulation is arbitrarily superimposable contrary to what is normally argued.

4.4 Surface Compositionality

We have seen above that the vacuity proofs of compositionality use rather nonstandard functions on strings. It seems that one can rule out many of these examples by requiring that the functions be well-behaved. The question is what kinds of functions are well-behaved from a linguistic perspective. A similar problem is created by the fact that Montague says very little about the identity of the ambiguation relation R. Clearly, it is not meant to denote just any relation.

Hausser coined in [Hausser, 1984] the expression 'surface compositionality'. This can be stated as follows. The lexicon contains basic expressions, and every complex expression is made from basic expressions through concatenation. Since constituents may be discontinuous, this boils down to the following: basic expressions are tuples of strings, and modes can only concatenate these parts to form the parts of the tuple. Whether or not duplication is allowed is not entirely clear, but it seems that duplication is necessary. If expressions are strings then we can formulate the principle as follows. Consider the expressions to be members of the algebra of strings $\langle A^*, \cdot \rangle$, with A the alphabet, and $\cdot$ the operation of concatenation. A *term* $t(x_1, \cdots, x_n)$ is as usual a well-formed expression made from the variables using $\cdot$ and no constants. A *term function* is a function that is the extension of a term (see [Burris and Sankappanavar, 1981]).

> **Surface Compositionality.** For every mode f the function f^μ is a term function of string algebra.

The formal details are worked out in [Kracht, 2003]. Hausser notes that Montague departs from this requirement in two ways. One is the use of syncategorematic expressions, and the other the use of deletion. Here is an example, the derivation of /`every man such that he sees Mary smiles`/, which reveals this.

(17)
1. /`man`/, basic expression
2. /`see`/, basic expression
3. /`Mary`/, basic expression
4. /`see Mary`/, from 2. and 3. using S5
5. /$\mathtt{he}_1$/, basic expression
6. /$\mathtt{he}_1$ `sees Mary`/, from 4. and 5. using S4
7. /`such that` $\mathtt{he}_1$ `sees Mary`/, from 6. using $S3_1$
8. /`every man such that he sees Mary`/, from 1. and 7. using S2
9. /`smiles`/, basic expression
10. /`every man such that he sees Mary smiles`/, from 8. and 9. using S4

Here, the operation F5 underlying S5 is concatenation, F4 is similar, only that the verb form of the second string is replaced by its third singular form. The operation underlying $S3_n$ is $F_{3,n}$, where $F_{3,n}(\vec{x}, \vec{y})$ is /$\vec{x}$ `such that` $\vec{z}$/, where $\vec{z}$ results from $\vec{y}$ by replacing every occurrence of $\mathtt{he}_n$/$\mathtt{him}_n$ by `he`/`she`/`it` and `him`/`her`/`it` (where the form is chosen according to some syntactic condition). A number of steps are dubious from the standpoint of surface compositionality. For example, the words /`every`/, /`such`/ and /`that`/ are not constituents of the sentence, not even parts. They are introduced by the rules. Second, the numbers subscripted to the pronouns, $_1$, are deleted, and sometimes also the pronouns themselves. This means that they are not part of the surface string.

The use of syncategorematic expressions is mostly unproblematic. We can at no cost introduce basic expressions of the desired kind and formulate a corresponding semantics. This would require the introduction of quantifiers /$\mathtt{every}_n$/, analogous to $\forall x_n$ in predicate logic. Also, empty pronouns have been argued for in many places, most prominently Government and Binding Theory. However, from the standpoint of Surface Compositionality the overt reflex of /$\mathtt{he}_n$/ is the empty word, and thus we can have only a single such pronoun. There are essentially two solutions to this problem. One is to renounce the use of free variables altogether. This is the route that P. Jacobson has taken, see [Jacobson, 1999]. Another is to face the use of free variables head on. We shall discuss this problem below, after we have discussed the development of DRT.

4.5 DRT

Discourse Representation Theory (DRT) presented a challenge to Montague Grammar. If the interpretation of a sentence is a proposition then reference to objects in that sentence should be impossible, contrary to fact.

(18) `Some man walks. He talks.`

This is the problem that is raised in [Kamp, 1981]. The theory proposed in that paper and developed further in [Kamp and Reyle, 1993] (among much other work) is that of partial maps into a model. Like in Montague Grammar, pronouns carry indices, so what gets interpreted is not (18) but (19).

(19) $\mathtt{Some}_1$ $\mathtt{man}_1$ $\mathtt{walks}_1$. $\mathtt{He}_1$ $\mathtt{talks}_1$.

The second sentence is interpreted against a partial map that makes the first true. This is a partial map β that sends x_1 to some man that walks. A Discourse Representation Structure (DRS) is a pair $D = [V : \Delta]$, where V is a finite set of variables, and Δ a set of formulae or DRSs. There are various constructors, such as a binary constructor $\Rightarrow$ to create complex DRSs. A partial function β makes D true if there is a V-variant β' of β such that all clauses of Δ are true. Here a V-variant β' of β is a partial map such that if $x \notin V$ then β' is defined on x if and only if β is, and they have the same value; and if $x \in V$ then β' is defined on x even if β is not (no condition on its value). A formula is true under a partial map if all variables are assigned a value and the formula is true in the standard sense. $[V : \Delta] \Rightarrow [W : \Sigma]$ is true under β if for every V-variant β' of β that makes Δ true there is a W-variant that makes Σ true.

Unlike standard quantification where the side effect of the quantification is removed, the assignment is kept and the second sentence is interpreted using that assignment. It is true therefore if $\beta(x_1)$ also talks.

DRT was originally thought to exemplify the noncompositional nature of natural language meanings. Yet, later [Zeevat, 1989] proposed a compositional interpretation. Basically, a compositional account is possible in the same way as it can be given in predicate logic: the meaning of a formula is not truth under an assignment, rather, it is a set of assignments. For then the interpretation of a quantifier, say $\exists x_n$, can be given as follows:

(20) $[(\exists x_n)\varphi] = \{\beta : \text{there is } \beta' \text{ with } \beta' \sim_n \beta \text{ and } \beta' \in \varphi\}$

Define the map C_n on sets of assignments by

(21) $C_n(A) := \{\beta : \text{there is } \beta' \text{ with } \beta' \sim_n \beta \text{ and } \beta' \in A\}$

Then

(22) $[(\exists x_n)\varphi] = C_n([\varphi])$

This allows to interpret λ-abstraction as well. Zeevat notes however that the arrow $\Rightarrow$ is not interpreted properly. His own solution is to take as meaning the pair $\langle V, C\rangle$, where C is the set of satisfying assignments and V the set of main discourse referents.

4.6 The Problem of Variable Names

The solution just discussed hurts itself against the Principle of Surface Compositionality. For to assume that /$\texttt{some}_1$/ quantifies over x_1 and /$\texttt{some}_2$/ over x_2 is to assume that the indices are part of the surface strings, which they clearly are not. On the other hand, any occurrence of /`some`/ can be seen (under an appropriate indexing) as an occurrence of /$\texttt{some}_n$/ for any given n. This is because the actual indices distributed by the grammar may vary according to the context, and may be chosen arbitrarily subject only to the condition that different variables must bear different indices. This means that the meaning of /`some`/ is the disjunction of all meanings of /$\texttt{some}_n$/. This problem has been raised in [Vermeulen, 1995]. Vermeulen solves the problem of variable choice as follows. By default, two representations talk about different objects no matter whether they use the same name or not. There is a mechanism of assigning *names* to variables that allows to communicate between formulae the intention to regard two uses of same variable (not necessarily the same!) as taking about the same object. [Fine, 2003] discusses the same problem with respect to the semantics of ordinary predicate logic. Fine expands on this theme in [Fine, 2007]. In his words, there is no guarantee that the use of the same variable is meant to make reference to the same object. If they do, they are said to be *coordinated*. Coordination happens only under under restricted circumstances.

In the context of compositionality the question is this. Suppose we have two representations a and b, say in the form of two formulas $\varphi(\vec{x})$ and $\chi(\vec{y})$, and we wish to "merge" them. How shall we rename the variables of, say, χ in order to perform the correct coordination?

In this formulation we can assume that a single formula is well coordinated in the sense that variables with different names can assume different values, while variables with the same name must assume the same value. The principle of independence from names can be formulated as follows.

Alphabetic Innocence. Two formulas φ and χ represent the same meaning if there is an injective renaming s of the variables such that χ is the result of replacing each variable x by $s(x)$ in φ.

This principle has deep consequences. Consider the formula $\mathsf{see}'(x_0, x_1)$. Consider the renaming $s : x_0 \mapsto x_1, x_1 \mapsto x_0$. Then the result of applying this renaming is the formula $\mathsf{see}'(x_1, x_0)$. Thus, the meaning of the two formulae is the same. The meaning cannot be a relation in the standard sense, for then a relation must be the same as its converse. (Or, as [Williamson, 1985] and [Fine, 2000] claim, a relation *is* identical with its converse, and the standard positionalism is wrong-headed.)

Under such constraints compositionality seems hard to maintain. Indeed, it can be shown that predicate logic has no compositional semantics ([Kracht, 2007b]). However, any finite variable fragment does. Also, natural language semantics becomes something of a different enterprise. As is shown in [Kracht, 2007a], for example, it can under these conditions be shown that Dutch is not strongly context free even if it is weakly context free.

4.7 Meaning Postulates and Logical Form

A somewhat disregarded theme in Montague Grammar is the use of meaning postulates. A discussion is found in [Dowty, 1979] and [Zimmermann, 1999]. Meaning postulates go back at least to Carnap. Montague introduces them for a specific purpose. The strategy of raising to the worst case introduces too many degrees of freedom. For example, if names are now on a par with proper nouns, their interpretation can be any set of individual concepts. But that is not what names are supposed to denote; they are more specific. They are such sets of individual concepts that are true of a single object. Also, Montague noted the following problematic inference.

(23)
```
   The temperature rises.
   The temperature is ninety degrees.
   ----------------------------------
∴ Ninety rises.
```

A proper analysis must take into account that /`rise`/ is a property not of individuals but of individual concepts. A temperature can rise since it is a function from worlds to numbers, a particular number cannot rise. Montague therefore opted to intensionalise all arguments of a verb. This excludes the dubious inference, but it also excludes inferences that are valid no matter what.

(24)
`The president talks.`
`Nicolas Sarkozy is the president.`
$\therefore$ `Nicolas Sarkozy talks.`

Hence the solution is to add a meaning postulate to the effect that /`talks`/ is transparent with respect to its subject.

In its strictest definition a meaning postulate is a decomposition of a primitive expression into simpler ones (such as Montague's decomposition of "seek" into "try to find"). The virtue of such a decomposition is not apparent at first sight. However, as rules of quantification allow intermediate scopes, it is not the same to have a single primitive expression and two have a composition of several of them. Also, as is emphasised in Generative Semantics, the choice of primitives may reveal something about the underlying semantic regularities of a language. [Dowty, 1979] argues in a similar way.

At the other extreme, meaning postulates are any formula constraining the meaning of some primitive. This means that a meaning postulate is nothing but an axiom in the ordinary sense. It is however not clear whether this should be held against Montague Grammar. Fpr it is clear that even an analysis in terms of a logical language relies ultimately on axioms to secure a minimum of material content to its symbols, logical or not.

5 Conclusion

Montague Grammar has inspired several generations of formal semanticists. It has paved the way to a precise formulation of semantic problems and solutions. Montague has shown that it is possible to do highly rigorous work and yet make substantial progress at the same time. What it certainly is not, however, is the last word on matters. Especially when it comes to compositionality there is no consensus whether Montague has supplied a fully compositional approach. This has nothing to do with a lack of precision; it has more to do with the question whether the abstract formulation is a good rendering of our initial intuitions. In many ways, Montague looks more like a technician than a theoretician; he prefers something that works over something that has intrinsic virtues. Forty years on, the ideas have been substantially modified. We seem to have a much more profound notion of what is a compositional semantics.

References

[Bach and Wheeler, 1983] Emmon Bach and Deirdre Wheeler. Montague Phonology: A First Approximation. In W. Chao and D. Wheeler, editors, *Problems of Linguistic Metatheory*, number 7 in University of Massachussetts Occasional Papers. 1983.

[Burris and Sankappanavar, 1981] Stanley Burris and H. P. Sankappanavar. *A Course in Universal Algebra*. Number 78 in Graduate Texts in Mathematics. Springer, 1981.

[Calcagno, 1995] Mike Calcagno. A Sign–Based Extension to the Lambek Calculus for Discontinuous Constituents. *Bulletin of the IGPL*, 3:555 – 578, 1995.

[Cooper, 1975] Robin Cooper. *Montague's Semantic Theory and Transformational Syntax*. PhD thesis, University of Massachussetts, Amherst, 1975.

[Cresswell, 1973] Max Cresswell. *Logics and Languages*. Methuen & Co., London, 1973.

[Davidson, 1967] Donald Davidson. The Logical Form of Action Sentences. In Nicholas Rescher, editor, *The Logic of Decision and Action*. University of Pittsburgh Press, Pittsburgh, 1967.

[de Groote, 2001] Philippe de Groote. Towards Abstract Categorial Grammars. In *Association for Computational Linguistics, 39th Annual Meeting and 10th Conference of the European Chapter*, pages 148 – 155, Toulouse, 2001.

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

[Dowty, 1979] David Dowty. *Word Meaning and Montague Grammar*. Reidel, Dordrecht, 1979.

[Fine, 2000] Kit Fine. Neutral relations. *The Philosophical Review*, 109:1 – 33, 2000.

[Fine, 2003] Kit Fine. The Role of Variables. *Journal of Philosophy*, 50:605–631, 2003.

[Fine, 2007] Kit Fine. *Semantic relationism*. Blackwell, London, 2007.

[Geach, 1972] Peter Geach. A Program for Syntax. In Donald Davidson and Gilbert Harman, editors, *Semantics for Natural Language*, number 40 in Synthese Library. Reidel, Dordrecht, 1972.

[Hausser, 1984] Roland R. Hausser. *Surface Compositional Grammar*. Wilhelm Finck Verlag, München, 1984.

[Heim and Kratzer, 1998] Irene Heim and Angelika Kratzer. *Semantics in Generative Grammar*. Blackwell Publishers, Oxford, 1998.

[Hendriks, 1993] Herman Hendriks. *Studied Flexibility. Categories and Types in Syntax and Semantics*. PhD thesis, Universiteit van Amsterdam, 1993.

[Hendriks, 2001] Herman Hendriks. Compositionality and Model–Theoretic Interpretation. *Journal of Logic, Language and Information*, 10:29 – 48, 2001.

[Jacobson, 1999] Pauline Jacobson. Towards a variable free semantics. *Linguistics and Philosophy*, 22:117–184, 1999.

[Janssen, 1983] Theo Janssen. *Foundations and applications of Montague Grammar*. PhD thesis, Universiteit van Amsterdam, 1983.

[Janssen, 1997] Theo Janssen. Compositionality. In Johan van Benthem and Alice ter Meulen, editors, *Handbook of Logic and Language*, pages 417 – 473. Elsevier, Amsterdam, 1997.

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

[Kamp, 1981] Hans Kamp. A theory of truth and semantic representation. In Jeroen Groenendijk, editor, *Formal methods in the study of language*. Mathematisch Centrum, 1981.

[Keenan and Faltz, 1985] Edward L. Keenan and Leonard L. Faltz. *Boolean Semantics for Natural Language*. Reidel, Dordrecht, 1985.

[Kobele, 2006] Greg Kobele. *Generating Copies: An Investigation into structural identity in language and grammar*. PhD thesis, Department of Linguistics, UCLA, 2006.

[Kracht, 2003] Marcus Kracht. *The Mathematics of Language*. Mouton de Gruyter, Berlin, 2003.

[Kracht, 2007a] Marcus Kracht. The Emergence of Syntactic Structure. *Linguistics and Philosophy*, 30:47–95, 2007.

[Kracht, 2007b] Marcus Kracht. Lectures on interpreted languages and compositionality. Manuscript, UCLA, 2007.

[Montague, 1970a] Richard Montague. English as a Formal Language. In R. Visentini, editor, *Linguaggi nella Societa a nella Tecnica*. 1970.

[Montague, 1970b] Richard Montague. Universal grammar. *Theoria*, 36:373–298, 1970.

[Montague, 1973] Richard Montague. The Proper Treatment of Quantification in Ordinary English. In J. Hintikka, J. Moravcsik, and P. Suppes, editors, *Approaches to Natural Language*. Reidel, Dordrecht, 1973.

[Montague, 1974] Richard Montague. *Formal Philosophy: Selected Papers of Richard Montague*. Yale University Press, New Haven and London, 1974. edited by Richmond H. Thomason.

[Moortgat, 1993] Michael Moortgat. Generalized quantifiers and discontinuous type constructors. In W. Sijtsma and A. van Horck, editors, *Discontinuous Constituency*. Mouton de Gruyter, Berlin, 1993.

[Morrill, 1994] Glyn V. Morrill. *Type Logical Grammar. Categorial Logic of Signs*. Kluwer Academic Publishers, Dordrecht, 1994.

[Muskens, 1995] Reinhard Muskens. *Meaning and Partiality*. CSLI, Stanford, 1995.

[Muskens, 2001] Reinhard Muskens. Lambda Grammars and the Syntax–Semantics Interface. In Rob van Rooy and Martin Stokhof, editors, *Proceedings of the Thirteenth Amsterdam Colloquium*, pages 150–155, 2001.

[Oehrle, 1988] Richard T. Oehrle. Multi–Dimensional Compositional Functions as a Basis for Grammatical Analysis. In Emmon Bach Richard T. Oehrle and Deirdre Wheeler, editors, *Categorial Grammars and Natural Language Structures*, pages 349 – 389. Reidel, Dordrecht, 1988.

[Partee, 1986] Barbara H. Partee. Noun Phrase Interpretation and Type-Shifting Rules. In Jeroen Groenendijk, D. de Jong, and Martin Stokhof, editors, *Discourse Representation Theory and the Theory of Generalized Quanifiers*, pages 115–143. Foris, 1986.

[Partee, 2004] Barbara H. Partee. *Compositionality in Formal Semantics*. Blackwell, Oxford, 2004.

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

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

[Westerståhl, 1998] Dag Westerståhl. On Mathematical Proofs of the Vacuity of Compositionality. *Linguistics and Philosophy*, 21:635 – 643, 1998.

[Williamson, 1985] Timothy Williamson. Converse relations. *The Philosophical Review*, 94:249 – 262, 1985.

[Zadrozny, 1994] Wlodek Zadrozny. From Compositional Semantics to Systematic Semantics. *Linguistics and Philosophy*, 17:329 – 342, 1994.

[Zeevat, 1989] Henk Zeevat. A compositional approach to Discourse Representation Theory. *Linguistics and Philosophy*, 12:95–131, 1989.

[Zimmermann, 1999] Thomas Ede Zimmermann. Meaning Postulates and the Model–Theoretic Approach to Natural Language Semantics. *Linguistics and Philosophy*, 22:529 – 561, 1999.