stichting
mathematisch
centrum

MC

AFDELING ZUIVERE WISKUNDE ZN 83/78 MEI
(DEPARTMENT OF PURE MATHEMATICS)

T.M.V. JANSSEN

# COMPOSITIONALITY AND THE FORM OF THE RULES IN MONTAGUE GRAMMAR

2e boerhaavestraat 49 amsterdam

Printed at the Mathematical Centre, 49, 2e Boerhaavestraat, Amsterdam.

The Mathematical Centre, founded the 11-th of February 1946, is a non-profit institution aiming at the promotion of pure mathematics and its applications. It is sponsored by the Netherlands Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O).

AMS(MOS) subject classification scheme: 02A05, 02C99, 68A30

# Compositionality and the form of the rules in Montague grammar

by

T.M.V. Janssen

ABSTRACT

Two fundamental principles concerning Montague grammar are proposed and a formalization of these principles is given. It is investigated which consequences these principles have for the possible forms of the rules in a Montague grammar.

KEY WORDS & PHRASES: *Montague grammar, Frege's principle*

## 1. THE PRINCIPLES

In the last years several extensions and variants of PTQ (MONTAGUE 1973) have been published. They deal with phenomena which are not treated in PTQ or are treated in a defective way. In order to do so, often new kinds of rules and new technical tools are incorporated in the system; a most interesting example is the attempt to make a synthesis of Montague grammar and transformational grammar (PARTEE 1973,1975). This growth of the kinds of rules is not restricted by PTQ since PTQ does not provide a description of what a possible rule is like. On the contrary, its syntactic rules suggest that any description of the desired effect by means of an English sentence is acceptable. This way of description has been used in semantics as well. Since this tool has a great expressive power, it is not surprising that the new proposed rules differ considerably.

Such an uncontrolled growth of the types of rules is for several reasons undesirable. The most important are probably the considerations of methodological nature. A standard definition provides, for instance, for a languange in which one can formulate observations, relations and generalizations and it provides a good starting point for formulating extensions, restrictions or deviations of the framework. Some objections of practical nature are as follows. If each "extension" uses a deviant framework, then one has to start each time anew in obtaining intuitions about the properties of the system and to check whether old knowledge still holds. If one uses a computer program then it means that one has to rewrite the program completely, probably using new data-types. As long as there is no standard, it is impossible to avoid ad-hoc programming. Objections of a linguistic nature are presented by PARTEE 1978. She wishes to restrict the concept of Montague grammar in order to come to a characterization of the possible natural languages. Finally, an important justification for this research can be found in the interesting results obtained by it.

Our investigations concerning the possible kinds of rules in a Montague grammar will be based on two principles. These principles have a simple and natural formulation, they are intuitively very appealing and I expect that hardly anybody will disagree with them. The priciples are:

SEMANTIC COMPOSITIONALITY PRINCIPLE:

*The meaning of a compound expression is built up from the meanings of its constituent parts.*

SYNTACTIC COMPOSITIONALITY PRINCIPLE:

*Each syntactic rule operates on the well-formed expressions of specified categories in order to produce a well-formed expression of a specified category.*

The semantic compositionality principle has a long history. It goes back to Tarski and Frege and therefore it is sometimes called the Fregean principle. This principle gave rise to the development of model-theoretic semantics for logic. The principle is fundamental for the Montague style of dealing with semantics and it is widely known and generally accepted among people working in Montague grammar. The syntactic compositionality expresses the way in which PTQ works and the way in which the syntax of much formal grammars is organized. The formulation of the syntactic compositionality principle is almost the same as the formulation of the well-formedness constraint of PARTEE 1978. We have chosen another name for the principle because we will give it a different interpretation from what Partee does; the name well-formedness constraint will be used for an interpretation which is more according to her ideas.

As a direct application of the well-formedness constraint Partee considers a rule which constructs adnominal adjectives from relative clauses. Its syntactic function $F_i$ has the effect that:

$F_i$*(immigrant who is recent) = recent immigrant.*

The input for this rule is an ill-formed expression *(immigrant who is recent)* and she judges that therefore such a rule is interdicted by the well-formedness constraint. This argmentation is, in my opinion, based upon a confusion between a general definition of a possible grammatical rule, actual examples of such a rule and the notion adequacy of a grammar.

On the one side one may consider an abstract definition of the kind of grammars one wishes to use. This definition contains a definition of what the possible rules of a grammar are. A simple example would be the definition of a context-free rewriting rule. It is a natural requirement that on the basis of this definition one is able to decide whether a certain text describes a rule which satisfies the definition. If there is no effective way to obtain a definite answer on this question, such a definition is useless. The definition of the grammar also contains a description of how the rules should be used to produce strings over some alphabet. These strings are called the *well-formed expressions* over this alphabet generated by the grammar. A subset of the well-formed expressions is called the *generated language*. The generated language of the PTQ grammar for a fragment of English consists of the generated expressions of the category t. The language of intensional logic consist of all expressions generated by its grammar. According to the PTQ grammar for English the expressions *love* $him_1$ is well-formed whereas *love her* is not well-formed since this expression is not generated by the grammar.

On the other side one may consider some actual grammar satisfying the definition and some -in an other way defined- language (for instance English). Then one may ask whether this language is the same as the language generated by the grammar; In other words one may ask whether the grammar is adequate for that language. One should be aware that adequacy of a grammar is completely different notion from the notion definition of a grammar. Grammars may exist without being adequate for some natural language: in the definition of a grammar adequacy plays no role.

Clearly Partee understands by a well-formed expression an expression occuring as subexpression in some English sentence. Her well-formedness constraint states that all expressions produced by the grammar must be well-formed expression in the above sense. This is a mixing up of definition and adequacy and makes her constraint to an unusable one. Suppose that one is presented a list of rules and one is asked whether this list constitutes according to her constraint a list or rules of a grammar for English. In order to answer this question one may start to produce some strings and ask each time whether it is an well-formed expression English. Suppose they are well-formed, then one cannot conclude that the list obeys the constraint since not all possible outcomes of the rules are considered. One has to try and try again, but never the definite answer "yes" can be given (related questions in formal language theory are known to be recursively unsolvable). It is of course possible for some special lists to obtain a definite answer (e.g. if the list consists of one non-recursive rule), but a constraint one can check only for some special lists and not for others is not a usuable constraint on the system. So the well-formedness constraint cannot be accepted as a formal constraint on the possible rules in a Montague grammar. It must be considered as expressing just some hope or desire: namely that an adequate grammar of English will be such that it contains no rule which produces *recent immigrant* from *immigrant who is recent*. In section 4 we will try to guarantee some of this desire by a formal constraint.

We have to say ourselves what we understand by the phrase "well-formed expression" in the syntactic compositionality principle. It is the other interpretation: the well-formed expressions are the expressions generated by the rules of the grammar. Any output of a syntactic rule is produced by that syntactic rule, so it is a wellformed expression. The phrase in the principle stating that the rules produce well-formed expressions is a pleonasm. The same holds for the inputs: the only possible expressions of specified categories are the expressions generated by the grammar. The word well-formed gives no restriction on the expressions, it does not contribute anything to the meaning of the syntactic compositionality principle and it may be ommited. In section 4 we will consider what is left of the principle.

Thus we observe that the interpretation of the syntactic compositionality principle is not uniquely determined. There occur so much vague, undefined words in the principles that they hardly say anything; everybody can find its own interpretation in them. Maybe this explains why these principles are so attractive and acceptable. In order to give substance to the principles, we must make the interpretation of several such words explicit. The guiding policy will be to give the principles as much content as possible, they should express sufficient and necessary conditions. It will appear that the compositionality principles together with the additional definitions give rise to interesting claims about the possible rules in a Montague grammar: several rules proposed in the literature are interdicted by the principles. If someone is not willing to accept some of the consequences of the principles, he may reject the principles, or he may give another interpretation for the vague words. We will consider just a formalization, although I consider it as a rather straightforward one.

## 2. SEMANTIC COMPOSITIONALITY

In this section we will make explicit what we understand by building up the meaning of a "compound expression" from the meanings of its constituent parts. By a compound expression we understand an expression that is produced by some syntactic rule from some input expression. It does not matter how complex or simple the expression is. By speaking about "constituent parts" we do not mean that the compound expression is divided into parts and that we combine the meanings of these parts. We allow that certain parts have no meaning in isolation; such expressions are called syncategorematic expressions (examples from PTQ: *every*, *such that*). Since they do not contribute a meaning from which we can build a new one, there is no use in considering them as constituted parts. We also allow that words from the constituent parts are disappeared in the compound expression (in PTQ: term-substitution). By the parts of a compound expression we understand the expressions which served as inputs for the syntactic rule according to which the compound expression is constructed. But knowing these inputs is not enough to determine in which way they are constituents (*John* and *love* $him_1$ can be combined in at least two ways). We have to know in which sequel they served as inputs and of which syntactic rule. Therefore, by "the meanings of its constituent parts" we understand a list of meanings of syntactic expressions together with the information in which sequel they served as inputs and of which syntactic rule. This is the information which is available for constructing the meaning of the compound expression. At the one hand we may use all this information. For each syntactic rule we may combine the meanings of the constituent parts in another way. Therefore we will have for each syntactic rule a separate semantic rule. On the other hand we may use only this information. If the principle would be understood as stating that you may use this information, but also other information if you wish so, then the principle would not state the whole truth and it would become a hollow phrase.

What does "meaning" mean? Let us consider the case of declarative sentences. According to PTQ, such a sentence is translated into an expression of intensional logic (henceforth IL). The interpretation of this expression with respect to some point of reference and some variable assignment yields a truth-value. This truth-value does not constitute the meaning of the sentence; meaning is more. The meaning of a sentence determines the circumstances under which the sentence is true or false. So knowing the meaning is (for all variable assignments) knowing for what kind of indices the interpretation yields true and for which false. Generally stated, it is knowing (for each variable assignment) a function from indices to the possible denotations of a certain type: it is knowing the intension. (The reader might consult LEWIS 1972 for a more extensive argumentation to this approach to semantics). So meaning is some element in the domains of the model. This element can be represented by some expression from IL. The interpretation of this formula with respect to a certain index gives us the value of the intension function for that index. By an expression representing a certain meaning we understand such an IL-formula. Notice that the translation function from PTQ yields such a meaning representating expression. One should be aware that such an expression is not identical with the meaning it represents. One and the same meaning can be represented by several expressions. Each of them is equally good in this respect.

The formalization of the concept meaning we considered above, it not the last and final answer. The intuitive concept meaning is rather vague, one may relate it with several phenomena, which are not covered by the above approach. In our formalization all tautologies have the same meaning, if one is interested to discriminate among them, another formalization is needed. (See LEWIS 1972). No extension of PTQ actually uses this formalization of meaning, in essence all have the same formalization: meaning is an intension.

What is allowed for building up new meaning? Meanings are functions, and we can do with them everything that can be done with functions. I see no argument to restrict our tools here. For building a new meaning from old meanings we allow every method which can be used to define a new function from old ones. We consider some examples.

1. If the meanings of the expressions $\phi$ and $\psi$ are functions yielding truth-values, then we may define a new function which yields true for a certain index if and only if both the meaning of $\phi$ and of $\psi$ yield true. Then the new meaning can be represented by $\phi \wedge \psi$.

2. Suppose that the interpretation of $\alpha$ may operate on the meaning of $\beta$, then we may define a new meaning as the intension yielding at each point of reference the result of this operation. This new meaning is represented by $\alpha(\hat{}\beta)$.

3. Suppose that the interpretation of $\eta$ yields a truth value and $\eta$ contains the free variable z then one may define as new meaning that function which yields true for an index if there is a variable assignment for the variable z such that the interpretation of $\eta$ yields true for that assignment. This new meaning can be represented by $\exists z\eta$.

One observes that in all the cases above, the interpretation of the new meaning consists of a single IL-formula containing the representations of the old meanings. These occur unchanged and identifiable in the representation of the new meaning: the representation of the first constituent part occurs unchanged at a specified position, the second at another position and so on. If we would allow that the representations of the old meanings become changed, we would define an operation on representations. This does not always define an operation on the meanings they represent (see also section 3). We make the semantic compositionality principle operational by requiring that *we may only define operations on meanings by a providing a single formula from IL (or an appropriate extension thereof) which contains at specified positions the unchanged representations of the input meanings*. One might at first glance be tempted to think that this operational version is more restrictive than the original formalization we give of the principle. This is, however, not completely true; it is a restriction on the format in which operations on meanings can be represented. If someone considers IL as being too restrictive for his purposes, he may extend IL by new operators etc. The interpretation of such operators must recursively be defined as is done in PTQ for the usual ones ($\Box$,H,$\exists$,$\wedge$). The main advantage of requiring that the representation of an operation on meanings is presented by an IL expression is that this way of presenting guarantees that the compositionality principle is obeyed automatically.

We have explained rather extensively how we came to our formalization and operationalization of the semantic compositionality principle. I hope that no one will be surprised by it, we followed the obvious way from intuition towards a precise formalization. Some of the implications of this formalization have been stated by others, although not with the argumentation and coherence as we did. PARTEE 1973 states that the "the translation rule must be such that the translation of the input expression must occur intact in the translation of the output". PARTEE 1978 mentions as a "semantical constraint" that the meaning of a compound should be given by means of an IL formula. The more surprising is it, that several authors present extensions or variations of PTQ which do not obey the so fundamental principle of semantic compositionality. In most cases it is not the complexity of the problems which makes it difficult to obey the principle; in contrary, often it is not so difficult to provide a proposal which is in accordance with the principle. Probably one is not aware of the implications of the principle one accepts; therefore it seemed useful to present such an extensive argumentation. In section 3 we will consider some proposals which do violate the principle.

## 3. EXAMPLES CONCERNING SEMANTIC COMPOSITIONALITY

### 1. Shake John awake

DOWTY 1976 treats, among others, the semantics of factive constructions such as *shake John awake*. In order to do so, he extends the language of intensional logic with two operators: CAUSE and BECOME. Interesting for our discussion is his treatment of CAUSE. In order to define its in interpretation Dowty adds "to the semantic apparatus of PTQ a selection function f that assigns to each wff $\phi$ and each $i \in I$ a member $f(\phi,i)$ of I. [Intuitively $f(\phi,i)$ is to be that i' most like i with the (possible) exception that $\phi$ is the case [..]]". Then the interpretation of CAUSE reads:
"If $\phi,\psi \in ME$ then $(\phi \text{ CAUSE } \psi)^{A,i,j,g}$ is 1 if and only if $[\phi\wedge\psi]^{A,i,j,g}$ is 1 and $[\neg\psi]^{A,f(\neg\phi,i),j,g}$ is 1.

The function f is defined on IL-expressions and not on the interpretations of these expressions. As a consequence CAUSE is an operator on IL-expressions and not on the meanings they represent. This is illustrated as follows. The definition of f allows that for some $\phi,\eta,i$ holds that $f(\phi\wedge\eta,i) \neq f(\eta\wedge\phi,i)$. So it may be the case that $\neg(\phi\wedge\eta) \text{ CAUSE } \psi^{A,i,j,g}$ yields 1 whereas $\neg(\eta\wedge\phi) \text{ CAUSE } \psi^{A,i,j,g}$ yields 0. So the compositionality principle is not obeyed and, moreover, the implications of this CAUSE are incorrect. A correction is possible by taking as domain for f the intensions of formulas: f assigns to each $d \in D_{<s,t>}$ and $i \in I$ a member $f(d,i) \in I$. Then a situation as described above is automatically excluded. The interpretation of CAUSE now becomes

"If $\phi,\psi \in ME_t$ then $\phi$ CAUSE $\psi^{A,i,j,g}$ is 1 if and only if $[\phi \wedge \psi]^{A,i,j,g}$ is 1 and $[\neg\psi]^{A,\underline{i},j,g}$ where $\underline{i} = f((\hat{}\neg\psi)^{A,i,j,g},i)$.

2. Horse Cannonero

DELACRUZ (1976) considers expressions like *the horse Cannonero*. Such expressions belong to a category $\bar{T}$ and they are generated by the following rule:

S3.1 If $\alpha \in B_T$ and $\zeta \in B_{CN}$ then $F_{21}(\zeta,\alpha)$ $P_{\bar{T}}$, provided that whenever $\zeta$ is of the form $he_n$, $F_{21}(\zeta,\alpha) = \alpha$; otherwise $F_{21}(\zeta,\alpha)$ = the $\zeta$ $\alpha$.

Examples:

$F_{21}$*(horse, Cannonero)* = *the horse Cannonero*

$F_{21}$*(horse, he$_1$)* = *he$_1$*

Translation rule:

T3.1 If $\alpha \in B_t$, $\zeta \in B_{CN}$ and $\alpha,\zeta$ translate into $\alpha',\zeta'$ respectively, then $F_{21}(\zeta,\alpha)$ translates into $\alpha'$ if $\alpha$ is of the form $he_n$; otherwise $F_{21}(\zeta,\alpha)$ translates into

(1) $\lambda P\exists y[\forall x[[\zeta'(x) \wedge \lambda \mathcal{P}\lambda z\mathcal{P}\{\hat{}\lambda x[{}^{\vee}z={}^{\vee}x]\}(\hat{}\alpha')(x)] \leftrightarrow x=y] \wedge P\{y\}]$

Translation rule T3.1 refers to the syntactic form of the input expressions of the syntactic rule. This means that in order to obtain the representation of the compound expression we need more than only the meanings of the inputs for the syntactic rule and the information which rule is used. So T3.1 violates the formalization of the semantic compositionality principle. The correction of this rule can be provided for in the syntax. The phenomenon considered by Delacruz provides evidence that among the terms we should distinguish syntactically Proper names and indexed pronouns and ask in S3.1 just for a proper name as input. Notice that the formula Delacruz presets is not the simplest one. I would prefer

(2) $\lambda P\exists y\forall x[\zeta'(x) \wedge \alpha'(\hat{}\lambda z[{}^{\vee}x={}^{\vee}z]) \leftrightarrow x=y] \wedge P\{y\}]$.

3. Easy to please

This example concerns a rule which is so close to a correct formulation that I would not like to call it a violation of the principle; is rather a (illustrative) slip of the pen. The main reason for mentioning it, is that we will use it in the discussion of the syntactic compositionality principle. We consider the following rule from PARTEE 1973.

Derived verb phrase rule:

If $\phi \in P_t$ and $\phi$ has the form ${}_t[{}_T[he_i]_{IV}[\alpha]]$, then $F_{104}(\phi) \in P_{IV}$, where $F_{104}(\phi) = \alpha'$, and $\alpha'$ comes from $\alpha$ by replacing each occurence of $he_i$, $him_i$, $him_i self$ by $he^*$, $him^*$ $him^*self$ respectively.

Examples:

$F_{104}$*(he$_1$ sees him$_1$self) = see him*self*

$F_{104}$*(he$_7$ is easy to please) = be easy to please.*

Translation rule

If $\phi \in P_t$ and $\phi$ translates into $\phi'$, then $F_{104}(\phi)$ translates into $\lambda x_i \phi'$.

From the formulation of the translation rule it is not as evident as in the previous example that the translation rule uses syntactic information. In order to decide what the actual translation is ($\lambda x_1 \phi$ or $\lambda x_2 \phi$ or ...) one needs to know the index of the first word of $\phi$. The correction of this rule rather simple, in analogy of term-substitution in PTQ we give the syntactic operation an index as parameter: so $F_{104}$ is replaced by $F_{104,i}$. In a later paper (PARTEE 1977) she corrected the rule in this way.

4. John who runs

BARTSCH 1976 and BARTSCH 1978 considers term phrases containing non-restrictive relative clauses. Such expressions are produced from a term and a sentence by the following rule (BARTSCH 1978)

S4. If $\alpha$ is a term and $\beta$ a relative sentence, then $\beta(\alpha)$ is a term. [...]

The corresponding translation rule reads

T4. If $\alpha'$ is the translation of the term $\alpha$ and $RELT(\lambda x\ \beta'(x))$ is the translation of the relative clause $\beta$ from S4, then $(RELT(\lambda x\ \beta'(x)))(\alpha')$ is the translation of $\beta(\alpha)$, and for all terms $\alpha$ with $\alpha' = \lambda P(..P(\nu)...)$ we have: $(RELT(\lambda x\ \beta'(x)))(\lambda P(...P(\nu)...)) = \lambda P(...\beta'(\nu)\ \&\ P(\nu)...)$.

Take as the representation for the meaning of the term *every man*

(3) $\lambda P\forall v[\text{man}'(v) \rightarrow P(v)]$.

If we combine (3) with the translation of some relative clause, the effect of T4 is well-defined. We might also consider another representation for the meaning of *every man*. Let Q be a variable of the same type as P and in (1) and Let $\mathcal{R}$ be a variable of the same type as the translation of term. Now the effect of T4 is not defined for:

(4) $\lambda Q\forall v[\lambda\mathcal{R}[\mathcal{R}(\text{man}') \rightarrow \mathcal{R}(Q)]\ (\lambda P\ P(v))]$.

A reaction on these objections against rules like T4 might be that one adds to the rule a clause stating that if the input formula is not in the required format, it must be reduced to that format. This a is a very simplified formulation of complex way of defining function between meanings. In order to define functions, one has to fulfill the following three requirements.

1. One has to describe exactly for which representations one will define the function.

2. One has to define for all expressions in the subset what the effect of the function is.

3. I consider two alternatives.

3a. One has to prove that each meaning for which we wish to define the function has precisely one representation in the subset defined in 1. (It lies at hand to prove this by providing an algorithm which transforms a given expression into one in the subset. This can, however, not be done since it would bring us in conflict with the undecidability of IL).

3b. One has to prove that each meaning for which we wish to define the function, has at least one representation in the subset and moreover that the result of applying the function to two different representations of the same meaning yields the same result.

Rule T4 does not fulfill the requirements. In general it is a complicated and extensive task to define a function between meaning by defining a function for specially selected representations. It can probably only be done in practice, if one considers a situation with a special structure in which all the proofs become drastically simplified. But if the situation is such a special one, one may expect that the same effect can be obtained in a more direct way as is demonstrated below.

The idea behind our reformulation is that the effect of replacing $P(\nu)$ by $\beta'(\nu)$ & $P(\nu)$ can be obtained by giving $\lambda z[\beta'(z)$ & $P(z)]$ as argument of $\lambda P[\ldots P(\nu)\ldots]$. We must take care of the binding of the variable and thus we come to a version of T4 which is in accordance with our formalization and operationalization of the semantic compositionality principle:

T4'. Let $\alpha'$ be the translation of the term $\alpha$ and $\gamma'$ the translation of the relative clause $\gamma$. Then the translation of the compound expression $\gamma(\alpha)$ is:

(5) $\lambda Q(\alpha'(\lambda z[\gamma'(z) \;\&\; Q(z)]))$

One observes that it is not needed to define the intended function from T4 along the laboured way of defining a mapping on special selected representations. The formulation of T4' is more exact and more simple than the formulation in T4.

One might take instead of (5) a more complex representation in which the translation of the term is operand rather than operator. This gives

(6) $\lambda \mathcal{R}[\lambda Q(\mathcal{R}(\lambda z[\gamma'(z) \;\&\; Q(z)]))](\alpha')$.

It is interesting to compare (8) with the expression RODMAN 1976 gives for the nonrestrictive relative clause:

(7) $\lambda \mathcal{P}[\lambda \mathcal{Q}\ \mathcal{P}\{{}^\wedge\lambda x_n[\gamma \;\&\; \mathcal{Q}\{x_n\}]\}]\ ({}^\wedge\alpha')$.

It turns out that their approaches are semantically in essence the same. This illustrates the use of writing semantic functions in the same format, from the formulation in T4 the close relationship could not be observed.

5. a[p][q] =: y

JANSSEN & van EMDE BOAS 1977 present a Montague grammar for the syntax and semantics of the assignment statement in the programming language ALGOL 68. They treat the semantics of an assignment to arrays of dimension n by reducing it to the case of dimension n-1. Unfortunately this approach gives rise to problems concerning semantic compositionality principle. Adopting Lewis formalization of meaning (section 2) would make it possible to save the principle. This is, however, a rather sneaky escape since in the context of programming languages all tautologies do have the same meaning. Although the compositionality principle is not standard among computer scientist, the authors prefer to obey it. So the authors have to give up their claim that they can treat the semantics of assignments to arrays without the need of a separate rule for each dimension. Giving up this claim is not too hard: there remain enough reasons for preferring their Montague-still approach. It is the only known semantic treatment of pointers and it is a proposal which (now) obeys the semantic compositionality principle whereas several other proposals do not.

## 4. SYNTACTIC COMPOSITIONALITY

We will interprete the syntactic compositionality principle (just as we did for the semantic principle), as giving a necessary and sufficient condition. It is not so surprising that a rule may operate on expressions of specified categories. The interesting aspect of the principle is that it states that this is also sufficient. Once the input expressions of specified categories are available, the rule can be applied. One does not need to know in which larger expression context the expression will be used. If several rules can be applied, then they are equally possible: there is no prescribed order among them. The derivational history has no influence on the question whether the rule applies or not. Even the actual form of the input expression is not of importance: if expressions of the required categories are available, the rule always is applicable. *The syntactic rules must be total rules!*

There are several arguments for interpreting the syntactic compositionality principle in this way. The most important one is that the requirement of total rules gives rise, in combination with the semantic compositionality principle, to several important and attractive consequences concerning the form of the rules. One consequence will be discussed below others will be discussed in the next sections. Another argument is that our interpretation of the syntactic compositionality principle expresses the way in which much formal grammars work, for instance PTQ has total rules. It is moreover an attractive and elegant principle because of its analogy with (our formalization of) the semantic compositionality principle. Finally, there is a practical motivation for total rules. Total rules are easier to understand and can easier be handled by a computer program.

The first consequence of having total concerns the "un-well-formed" expressions. Suppose the grammar contains a rule $S_i$ of which the syntactic operation $F_i$ has the following effect:

$F_i$*(immigrant who is recent) = recent immigrant.*

So the rule operates on a common noun-phrase which, according to rule S3.1, must be constructed from the common noun *immigrant* and the sentence *he*$_1$ *is recent*. This sentence must come from the IV-phrase *be recent*. Since we require that the rules are total, we may also combine this IV-phrase with other term-phrases. So also the sentence *John is recent* is generated by this grammar, which is not a correct sentence of English. This example suggests that an adequate grammar for English cannot contain a rule which generates *recent immigrant* in the way rule $S_i$ does.

The above reasoning is not a mathematically proof that it is absolutely impossible that an adequate grammar for English contains the rule $S_i$. In fact it can be made possible by changing the PTQ rules and applying the following trick. We split each category in two new ones: one that may contain expressions such as *is recent* and one that may not contain them. So the grammar becomes rather complicated while a simple solution lies at hand. It is unlikely that someone will ever write down a grammar as sketched above. If some rule in the grammar introduces an "un-well-formed" expression, then it is due to the totalness of the rules, very difficult to get rid of that expression. So in a certain sense Partees well-formedness constraint is saved. The syntactic compositionality principle, with the interpretation of requiring total rules guides us towards a grammar which fulfills her desire.

## 5. TRANSFORMATIONS

In this section we will investigate the consequences of the two compositionality principles for the incorporation in Montague grammar of transformations as they are used in transformational grammars. Therefore we consider some fundamental aspects of such transformation. These are:

1. Transformations define mappings from trees to trees rather then from strings to strings.
2. If several transformations can be applied, then the order of application may be prescribed.
3. A transformation applies on one input tree.
4. A transformation imposes certain structural conditions on the analysis of the input tree.

In order to take care of the first point, it is required that a Montague grammar does not operate on plain strings but on trees or equivalently on labelled bracketings. Let us assume that the Montagues framework can be adapted in this way. This change of the system makes all the rules to rules which operate on trees, so in a certain sense all the rules in Montague grammar become transformations. In order to avoid confusion in terminology, we will use the name C-transformation ("Chomskyan") for transformations as they are used in transformational grammars. Once they are incorporated in Montague grammar they are called P-transformations ("phrase-structure"). The second point gives rise to problems concerning syntactic compositionality. It asks for more than only the category of the input expressions: it asks for information about the derivational history of the expressions. In a grammar which obeys the syntactic compositionality principle, there cannot be a prescribed order on the order of applications of the rules: only an implicit ording is possible. Notice that PARTEE 1978 comes, based upon the well-formedness constraint, to the same conclusion concerning this point. The third point is rather peculiar in the context of a Montague grammar. A syntactic rule in a Montague grammar may have any number of inputs. It seems rather artificial to incorporate transformation-like rules in Montague grammar, while at the same time restricting such rules to the case of rules with one input expression. The most problematic is the last point since it implies that C-transformations are partial rules. It is a very important aspect of C-transformations that they have conditions on the structural analysis of their inputs; this aspect makes them very attractive for prâctical use. It makes it possible to indicate what the relevant trees are without bothering about all details that are considered as being irrelevant. Evidently the aspect of structural conditions where for PARTEE 1978 the main reason for allowing partial rules.

In spite of the above considerations, we will continue to require total rules. We will incorporate C-transformations in Montague grammar by means of a slight reformulation that makes them total. The reader might be surprised by this reformulation and consider it at first glance as a sneaky trick used in order to obey the letter of the principle. This is not completely true: the reformulation expresses a different view on transformations and this has, in combination with the semantic compositionality principle, important consequences.

Let us demonstrate how the reformulation works. Suppose a C-transformation is given in the following format.

> If the input expression is of the category $C_1$ and it satisfies structural condition SC, then we may apply transformation T in order to obtain an expression of the category $C_2$ and else we may not apply T.

Its reformulation as a total rule has the following format:

> If the input expression is of the category $C_1$ then we may apply the transformation T'. Transformation T' reads as follows. If the input expression satisfies the structural condition SC, then apply transformation T and otherwise apply a specified do-nothing transformation.

By a do nothing transformation we understand a rule which gives one of the inputs unchanged as output. Notice that in this reformulation the do-nothing transformation reflects the essential difference with Montague rule descending from C-transformations. So the essential aspect of a P-transformation is the occurrence of a do-nothing transformation.

The reformulation expresses the conception that the P-transformation always applies if an expression of the required category is available and that this application always yields an output. For rules with one input the output may, under certain conditions, be that input itself. Since the input category and the output category are specified in the syntactic rule, this means that for a P-transformation with one input, the output has always the same category as the input. It is a nice coincidence that this is always the case for C-transformations. For transformations with more than input, the use of a do-nothing transformation implies that (at least) one of the inputs must have the same category as the input. This consequence disallows us to impose conditions on the inputs of a rule like S4 (which combines a term-phrase and an IV-phrase to a sentence).

The other consequence of having total rules is that the corresponding semantic rule always applies. We already observed that for every P-transformation it may under certain conditions be the case that one of the inputs is used as output. So in this case the meaning of the output expression is the same as the meaning of that input. We observed in section 3 that it is a consequence of the semantic compositionality principle that the semantic rule must be the same for all inputs: it may not depend on the structural analysis of the input expressions. So in all cases the meaning of the output of P-transformation must be the same as the meaning of the input. The translation rule corresponding with a P-transformation must be the identity mapping on one of its inputs. This means, for instance, that a variant of rule S14 from PTQ cannot impose conditions on its input expressions since the corresponding translation rule is not the identity on one of its inputs. If we restrict our attention to P-transformations with one input, we come to the following consequence. From the two compositionality principles it follows that *the transformations in a Montague grammar must be meaning preserving!* For transformational grammar this is a well-known requirement (e.g. PARTEE 1971). Some examples in the literature of transformations which are not meaning-preserving, will be considered in the next section.

## 6. EXAMPLES CONCERNING SYNTACTIC COMPOSITIONALITY

1. $He_1$ is loved

PARTEE 1973 considers the C-transformation Passive Agent Deletion. An example is:

$F_{102}$(*he*$_1$ *is loved by him*$_3$) = *he*$_1$ *is loved*

Translation: If $\phi \in P_t$ and $\phi$ translates into $\phi'$, then $F_{102}(\phi)$ translates into $\exists x_j \phi'$.

At the one hand this transformation applies only to input trees of a special form. At the other hand the translation rule is not the identity mapping. This means that we cannot reformulate this transformation as a total rule. So the traditional way of dealing with agentless passive is disallowed by the compositionality principles. THOMASON 1976 provides rules for generating agentless passive by means of categorical rules and DOWTY 1978 does so for the case of double passives. So there is an alternative way available for dealing with agentless passive. When we follow this way, it is rather strange to generate passives which do contain an agent by means of transformations; they also can be generated according to the rules of Thomason and Dowty.

2. Mary shakes John awake again

In section 3 example 1 we considered some semantic aspects of the proposals of DOWTY 1976 concerning the treatment of factives. Now we will consider some syntactic aspects of course, in doing this we cannot neglect the semantic aspects of the rules. Dowty produces the factive sentence *Mary shakes John awake* from the term *Mary* and the IV-phrase *shake John awake*. This IV-phrase in its turn is obtained from the TV-phrase *shake awake*. The first rule he presents for generating this TV-phrase is as follows.

S30 If $\alpha \in P_{IV}$ and $\phi \in P_t$ and $\phi$ has the form *he$_n$* is $\gamma$ then $F_{30,n}(\alpha,\phi) \in P_{TV}$ where $F_{30,n}(\alpha,\phi) = \alpha\phi$.

An example is:

$F_{30,1}$(*shake, he$_1$ is awake*) = *shake awake*.

The corresponding translation rule reads: If $\alpha$ translates into $\alpha'$ and $\phi$ translates into $\phi'$ then $F_{30,n}(\alpha,\beta)$ translates into

$$\lambda P\lambda x P\{{}^\wedge\lambda y\, [\alpha'(x,{}^\wedge\lambda P[P\{y\}])\ \mathit{CAUSE}\ [\mathit{BECOME}\ [\phi']]]\}.$$

Again this rule is a partial rule which is not meaning preserving. So we have to find another approach; can the above result be obtained by means of a total rule? For generating expressions like *shake awake* one only needs an adjective and a TV-phrase. So it lies at hand to try the following rule:

$S_{930}$ If $\alpha \in P_{TV}$ and $\beta \in P_{adj}$ then $F_{930}(\alpha,\beta) \in P_{TV}$ where $F_{930}(\alpha,\beta) = \alpha\beta$.

The corresponding translation rule would be

$T_{930}$ If $\alpha$ translates into $\alpha'$ and $\beta$ translates into $\beta'$, then $F_{930}(\alpha,\beta)$ translates into

$\lambda P \lambda x P\{{}^\wedge\lambda y[\alpha'(x, {}^\wedge\lambda P\, P\{y\})])$ *CAUSE* [*BECOME* $[\beta'(y)]]]\}$.

*Mary shakes John awake again* is ambiguous. On the one reading Mary has done it before, on the other John has been awake before. Dowty treats *again* as a sentence modifier and he needs two different sentences in the derivation in order to deal with the ambiguity. He probably starts his investigations along this line for historical reasons: it is the way in which such constructions are treated in generative semantics. By rule $S_{930}$ we are guided to another approach to this ambiguity. The one reading can be obtained by combining *again* with *Mary shakes John awake*, the other by combining it with *shake awake*. We do not go into details of this approach for the following reason. After considering several phenomena concerning factives, Dowty observes that his first approach is not completely adequate. He discusses extensively several alternatives and escapes. Finally he concludes "there would be no reason why we should not then take the step of simplifying rules S30-S32 drastically by omitting the intermediate stage in which a sentence is produced". He presents a rule which is identical with $S_{930}$ as the rule which he considers as the best one. So the syntactic compositionality principle has led us immediately to the solution which is the simplest and the best one. This example suggest that we might derive from the syntactic compositionality principle the advice "to ask for what you need and not for more".

3. Easy to see himself

In section   example   we considered the Derived Verb Phrase rule from PARTEE 1973. An example concerning this rule was

$F_{104}$(*he*$_1$ *sees him*$_1$*self*) = *see him***self*.

At the one hand the derived verb phrase rule is a partial rule, at the other hand its output belongs to a different category than its input. Therefore we cannot reformulate this rule as a total one using a do-nothing transformation. So we have to find another treatment for the cases where Partee uses this rule. Let us, in accordance with the syntactic compositionality principle, just ask for what we actually need and not for more. In the above example we only need a TV-phrase. So the syntactic rule becomes

$S_{902}$ If $\alpha \in P_{TV}$ then $F_{902}(\alpha) \in P_{IV}$ where $F_{902}(\alpha) = \alpha$ *him***self*

The corresponding translation rule reads:

$T_{902}$ If $\alpha$ translates into $\alpha'$, then $F_{902}(\alpha)$ translates into $\lambda x\ \alpha'(x, {}^\wedge\lambda P\ P\{x\})$.

PARTEE 1975 provides as an explicit argument for use of the Derived Verb Phrase rule the treatment of the sentence

(8) *John tries to see himself.*

If the derived verb phrase rule is used for (1) then the translation becomes:

(9) $try\ to'\ ({}^\wedge j, {}^\wedge\lambda x_3\ \ see'\ (x_3, {}^\wedge\lambda P\ P\{x_3\}))$.

Sentence (1) can also be generated according to the lines of PTQ. Then the translation is

(10) $try\ to'\ ({}^\wedge j, {}^\wedge see'\ (\lambda P\ P\{{}^\wedge j\}))$.

Partee gives some considerations why she might prefer (9) above (10). Rule $S_{902}$ is compatible with that opinion, it gives rise to a translation which is equivalent with (9). So we have found a simple, total rule.

Partee also presents the example

$F_{104}$(*he*$_7$ *is easy to please) = be easy to please.*

This is not treated by $S_{902}$. We search for a total rule and thus ask just for what we need for generating *easy to please*. We need an expression like *easy* and some TV-phrase. Let us, as Partee does, assume that we have a special category $\overline{ADJ}$ which contains easy, though etc. The resulting expression easy to please will be of the category ADJ'. Then we are guided to the following rule

$S_{903}$ If $\alpha \in P_{\overline{ADJ}}$ and $\beta \in P_{TV}$ then $F_{903}(\alpha,\beta) \in P_{ADJ}$ where $F_{903}(\alpha,\beta) = \alpha$ to $\beta$.

The translation of (this) *easy* must be such that it may be combined with a TV-translation in order to obtain an expression of the type of translations of adjectives. Then the translation rule reads

$T_{903}$ If $\alpha$ translates as $\alpha'$ and $\beta$ as $\beta'$ then $F_{903}(\alpha,\beta)$ translates into $\lambda x\ \alpha'(\lambda y \beta'(x, \lambda P\ P\{y\}))$.

Rule $S_{903}$ makes it possible to generate the expressions containing easy to please which we mentioned above. Unfortunately, Partee does provide an explicit semantics for the source of all her constructions (7), so it is difficult to compare the semantic consequences of $S_{903}$. I expect that she will finally end up with something close to the result of $T_{903}$. Concerning the syntax, it is demonstrated that our principles guides us to a much simpler treatment.

4. Comparatives

There are more situations where the traditional approach uses rules which ask for more input than actually is needed. We will make some remarks concerning the case of comparatives. An example is:

(11) *Fewer of the women came to the party than of the men.*

This sentence is derived from the (illformed) sentence:

(12) *Fewer of the women came to the party than of the men came to the party.*

This is in its turn derived from a source sentence like

(13) *Fewer of the women came to the party than x many of the men came to the party.*

PARTEE 1978 presents (11) as a difficult case for the well-formedness constraint since it uses an illformed source (12). For the syntactic compositionality principle the intermediate stage of (12) is undesirable since one can hardly get rid of this sentence. Moreover the rules relating comparatives with sources as (12) or (13) are rather complicated. Why following this approach? Maybe one judges that a source like (12) or (13) expresses the semantic content of the comparatives more completely than comparatives. Or one wishes to explain the semantic relations between comparatives by generating them from the same kind of source. In transformational grammar this might be valid arguments, no other formal tools than transformations are available. In Montague grammar there is semantic component in which such semantic relations can be formally expressed. So if we do not need such a source for syntactic reasons we may try another approach. The syntactic compositionality guides us to ask for just what we need. In order to make a comparison between two individuals concerning the amount of some property they have, we need two terms and a property. So we introduce a three place rule which has the effect that

$F_{910}$*(John, Bill, see women) = John sees more women than Bill.*

The semantic component has to express what is compared; the syntax needs no to do so.

Another rule might compare of two sets in which amount they are involved in a certain property.

$F_{911}$*(man, boy, come to the party) = fewer of the men come to the party than of the boys.*

One may also compare two individuals for two properties.

$F_{912}$*(John, Bill, see men, meet women) = John sees more men than Bill meets women.*

These examples do not provide for a treatment of the comparative. They just illustrate the kind of solution we would search for in accordance with the compositionality principles. The other examples in this section make us expect that a solution along these lines will be simpler than the traditional treatment.

REFERENCES


BARTSCH, R., 1976, *Syntax and Semantics of relative clauses*, in GROENENDIJK & STOKHOF 1976.

————————, 1978, *The syntax and semantics of subordinate clause constructions and pronominal reference*, in HENY & SCHNELLE 1978.

DOWTY, R.R., 1976, *Montague grammar and the lexical decomposition of Causative verbs*, in PARTEE 1976.

———————— , 1978, *Dative movement and Thomasons extensions of Montague grammar*, to appear.

DELACRUZ, E.B., 1976, *Factives and propositional level constructions in Montague grammar*, in PARTEE 1976.

GROENENDIJK, J. & M. STOKHOF (eds.), 1976, Amsterdam papers in formal grammar, vol. 1: *proceedings the Amsterdam Colloquium on Montage grammar and related topics*, Centrale Interfaculteit, University of Amsterdam.

HENY, F. & H. SCHNELLE, 1978, *Selections from the third Groningen Round Table Conference*. (to appear).

JANSSEN, T.M.V. & P. VAN EMDE BOAS, 1977, *On the proper treatment of referencing, dereferencing and assignment*, in Lecture notes in Computer Science 52, *Automata, Languages and programming*, A. SALOMAA & M. STEINBY (eds.), Springer.

LEWIS, D., 1972, *General Semantics*, in DAVIDSON & HARMAN (eds.), *Semantics of natural language*, Reidel. Reprinted in PARTEE 1976.

MONTAGUE, R., 1973, *The proper treatment of Qualification in ordinary English*, in J. HINTIKKA, J. MORAVCSIK & P. SUPPES (eds.), *Approaches to natural language*, Reidel. Reprinted in THOMASON 1974.

PARTEE, B., 1971, *On the requirement that transformations preserve meaning*, in C. FILLMORE & D.T. LANGENDOEN, *Studies in Linguistic Semantics*, Holt, Rinehart & Winston.

———, 1973, *Some transformational extensions of Montague grammar*, Journal of Philosophical Logic 2, 509-534, Reprinted in PARTEE 1976.

———, 1975, *Montague grammar and transformational grammar*, Linguistic Inquiry 6, 203-300.

———, (ed.), 1976, *Montague grammar*, Academic Press.

———, 1977, *John is easy to please*, in ZAMPOLLI (ed.), *Linguistic Structures Processing*, North Holland.

———, 1978, *Montague grammar and the well-formedness constraint.* (to appear)

RODMAN, R., 1976, *Scope phenomena, "movement transformations", and relative clauses*, in PARTEE 1976.

THOMASON, R.H. (ed.), 1974, *Formal Philosophy*, Selected papers of Richard Montague, Yale University Press.

———, *Some extensions of Montague grammar*, in PARTEE 1976.