stichting
mathematisch
centrum

MC

AFDELING ZUIVERE WISKUNDE
(DEPARTMENT OF PURE MATHEMATICS)

ZN 100/81 MAART

T.M.V. JANSSEN

MONTAGUE GRAMMAR AND FUNCTIONAL GRAMMAR

kruislaan 413 1098 SJ amsterdam

Printed at the Mathematical Centre, 413 Kruislaan, 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.).

1980 Mathematics subjects classification: 03A05, 03B40, 08A05

Montague grammar and functional grammar*)

by

Theo M.V. Janssen

ABSTRACT

This report presents an application of the algebraic framework of Montague grammar to Dik's functional grammar. It is demonstrated how in this way a modeltheoretic semantics can be associated with the structures produced in a functional grammar. In particular, it is shown how scope ambiguities can be dealt with.

KEY WORDS & PHRASES: *Montague grammar, functional grammar, natural language, algebraic semantics*

---

*) to appear in GLOT 3, special issue: 'Perspectives on functional grammar'

1. Aim

A functional grammar describes how completely filled predicate frames are produced and how from such structures sentences are obtained. So a functional grammar is a syntactic device which produces some language (i.e. the set of produced sentences). Semantic considerations evidently played a role in the design of the structures employed in functional grammar. However, these structures are not semantically interpreted: functional grammar does not assign a meaning to the produced structures. The aim of this article is to demonstrate how a functional grammar can be incorporated in a system which does deal with semantics as well. The system is an instance of the general framework described in Universal Grammar (Montague (1970)). Many technical details are taken from the Montague's most influencing article: 'On the proper treatment of quantification in ordinary English' (Montague (1973), henceforth PTQ).

The kind of semantics I will consider is known under names as logical semantics, model theoretic semantics or truth-functional semantics. I will not try to characterise notions such as 'meaning' or 'semantics'. I will rather follow the advice of Lewis who says: 'In order to say what a meaning is, we may first ask what a meaning does, and then find something that does that' (Lewis (1972:5)).

I wish to formalize those aspects of meaning that allow one to conclude from the truth of certain sentences the truth of certain other sentences. If the sentence *John walks and Mary talks* is true, then also the sentences *Mary talks* and *John walks* are true, and vice versa. And if it is true that

Every man walks, and that *John is a man,* then also *John walks* should be true. The meanings associated with FG-structures should formalize the information needed to obtain such conclusions. I do not aim at formalizing more aspects of meaning. So I am not interested, for example, in describing the precise differences between *walk* and *run,* or in explaining the deviant use of *run* in *running nose*. Probably the meanings of sentences considered in this article are not surprising for the reader. The interest of these meanings rather lies in the method by which they are obtained.

I will now consider some sentences of the fragment dealt with in this article. They form a subset of the fragment in Dik (1980), which is the source of the FG rules I will deal with.

(1) *John runs*

(2) *Every man runs.*

The predicate frames related to (1) and (2) are similar: the only difference is that in the first position another term is inserted. In analogy with (1), which expresses that the individual John has the property of running, one might expect that the meaning of (2) is that some object indicated by *every man* has the property of running. In discussing this idea, Lewis tells that in the dark ages of logic something like the following was told (Lewis (1972:35)). The phrase *every man* names an entity called 'the universal generic man', and this entity has just the properties which every man has. Then (2) would mean that the universal generic man has the property of running. Lewis illustrates what a strange entity this universal entity would be. Since not every man is black, or yellow, or of some other specific color, the universal generic man is not black, nor yellow, nor of any other color (yet neither is he colorless, since not every man - indeed not any - is colorless). So this cannot be a sound approach. A better description of the meaning of sentence (2) is that every entity that has the property of being a man, also has the property of running. This illustrates why the quantified term *every man* is treated in logic differently from the proper name *John*. But in functional grammar both terms are treated the same. Therefore sentence (2) constitutes a first basic problem for a semantics in functional grammar. Some kind of a method has to be found enabling one to obtain rather divergent meanings for closely related frames.

(3) *John loves a woman*

This sentence expresses that there is an entity that has the property of being a woman and that it stands in the relation of loving to the individual John. A sentence which has a predicate frame of the same structure as (3) is *John seeks a unicorn*. This sentence exhibits an ambiguity which is known as the 'de-dicto/de-re' ambiguity. On the de-re reading John is looking for a specific unicorn; for instance the one he got for his birthday. This reading implies that there exist unicorns. On the de-dicto reading, however, John will be happy with any unicorn he finds. This reading does not imply the existence of uniforns. The de-dicto/de-re ambiguity arises in connection with intensional verbs such as *seek*. Since such verbs are not in the fragment this ambiguity will not be treated explicitly.

(4) *Every man loves a woman.*

This sentence is ambigous. There are two possibilities concerning the relative scope of the quantified phrases. On the one reading there is a single woman loved by every man (say Mother Mary). On the other reading there is for every man some woman he loves (say his mother). A well known variant examplifying the same kind of ambiguity is *Every man in the room speaks two languages*. The ambiguity of (4) is an example from a large class of ambiguities: scope ambiguities of quantified term phrases. It is a pure semantic ambiguity in the sense that there is no syntactic motivation for relating (4) with two different structures. A functional grammar will provide as source for (4), on both readings, the same predication. Since it is not clear how to obtain two different readings given only one frame, we arrive at the second basic problem in providing a semantics for functional grammar. Is it possible in a functional grammar to deal with ambiguities which are of a purely semantical nature?

2. Method

The main principle of Montague's approach is the well-known principle of compositionality (or Frege's principle):

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

This principle is intuitively appealing and is widely accepted (although different people may have different interpretations of it). In case we deal with strings which are just 'glued' together, it is clear what the parts of the resulting expression are: the parts we 'glued' together. But what is to be understood by constituent parts in less elementary cases: if the grammar does not just glue strings together or does not produce strings? An example of the former situation arises when a grammar builds *take the apple away* out of *the apple* and *take away*. Functional grammar is an example of the latter case since it deals, in most stages of the production process, not with strings, but with frames. In order to be able to apply the principle in situations which go beyond the most elementary ones, the notions 'compound expression' and 'constituent part' have to be clarified.

By a *compound expression* I will understand an object which is produced by a syntactic rule; in a functional grammar these objects are partially or completely filled predicate frames, and the sentences which are finally obtained. The lexical predicate frames are considered as *uncompound* expressions. The syntactic rules tell us how expressions are built out of already built ones or basic ones. By the *constituent parts* of an expression A, which is built using rule R, I understand the expressions $B_1,\ldots,B_n$ out of which A has been built using R. The principle now requires that in case we build A out of $B_1,\ldots,B_n$ according to rule R, we have to build the meaning of A out of the meanings of $B_1,\ldots,B_n$ according to some operation on meanings associated with R. So far each construction step in the syntax, there has to be a corresponding step in the semantics: i.e. each syntactic rule has a semantic counterpart. It is a consequence of the principle of compositionality that not only the final outcome of the production process has a meaning, but that all syntactic expressions (structures) encountered in the production process have one as well.

The parallelism between syntactic and semantic rules has as a consequence that the meaning of a structure is determined by its syntactic construction process. This indicates what kind of answer will be given to the first problem of section 1. One and the same predicate frame can only obtain two different semantic interpretations if it is produced in the syntax in two different ways. A meaning assignment is then a function defined on the set of derivational histories, and each derivational history is related to just one meaning. Formulated in the terminology of Montague 1970: a meaning assignment is a function defined on a disambiguated language, i.e. the language of production trees, and formulated in the terminology of universal algebra (the branch of mathematics which deals with structures and their relations): meaning assignment is a homomorphism defined on some free algebra, i.e. the term-algebra of that language.

The expressions produced by the grammar will be related to a semantic domain. This domain may be viewed as a formal model of reality, in which, for instance, (formal representants of) individuals occur, and (formal versions of) their properties are modelled. Of certain kinds of expressions one might have an intuition how their meanings should be modelled, for instance, one is tempted to associate with a two place verb a two place relation. But we need a meaning for all kinds of expressions; for the frame related to *every man*, as well as the frame related to the positional satellite *in the garden*. Here the situation is less clear. What are reasonable candidates for their meanings? The answer to this question will mainly be based upon pure technical grounds. Semantical objects are chosen which make the system work; i.e. those objects which make that the correct relations between meanings of sentences can be laid. In this way the advice of Lewis quoted in section 1 is practised: do not ask what a meaning is, but ask what a meaning should do. Often one associates intuitively motivated interpretations with the in this way obtained meanings, for instance concerning a model of reality or concerning the processes in the human brains. The reader is invited to hold these interpretations as long as possible because it might help him to understand why the system works. But in case some unintuitive meanings are encountered, a safe interpretation of the semantic model is as a collection of abstract mathematical objects which have the required properties enabling us to reach our semantical aims.

The objects in our semantical model are rather abstract things, such as truthvalues, functions from pairs of truthvalues to truthvalues (e.g. conjunction), functions from objects called entities to truthvalues (predicates of entities) etc.. When we relate expressions to such semantic objects, we have to be able to describe them. Because it is not convenient to do this in some mathematical dialect of English, I will use a suitable formal language which is an extension of predicate logic. In order to avoid confusion between the expressions produced by a functional grammar and the expressions of the logic, I will call the latter formulas.

The logic I will use is not an intentional logic, but a variant of (extensional) type logic. Working with Montague grammar is often identified with using intensional logic, and indeed (nearly) all work in Montague grammar does use this kind of logic. Many classical semantic puzzles (e.g. de-dicto/de-re) involve intensional contexts and require intensional logic for their semantical treatment. But Montague's abstract framework does not a priori require to use an intensional logic: it is the intensional phenomena which require intensional logic. Since the sentences I will deal with do not create intensional contexts, there is no need to use intensional logic here. This simplifies the task of explaining the framework, and has as a consequence that the formulas (and semantic operations) we will meet, are somewhat different, and probably more familiar to most of you, than the ones occurring in PTQ.

I assume the reader to know how to interprete standard predicate logic, the interpretation of its extension will be given in the next section. So to represent an abstract object in the semantic model (a meaning) it will be sufficient to provide some formula of our logical language which represents that meaning. Notive that different formulas may represent one and the same meaning: e.g. $\phi \wedge \psi$ and $\psi \wedge \phi$. If we would like to replace some representation of some meaning by some other representation of the same meaning we are free to do so.

We need an extension of predicate logic in order to be able to represent the meanings of parts of sentences. For the sentences mentioned in section 1 as such, predicate logic suffices. The meanings of (1), (2) and (3) are represented in (5), (6) and (7) respectively, and the two of (4) are represented in (8) and (9).

(5) $run\ (john)$

(6) $\forall u\ [man(u) \rightarrow run(u)]$

(7) $\exists u\ [woman(u) \wedge love(john,u)]$

(8) $\forall v\ [man(v) \rightarrow \exists u\ [woman(u) \wedge love(v,u)]]$

(9) $\exists u\ [woman(u) \wedge \forall v\ [man(v) \rightarrow love(v,u)]]$

Operations on meanings can be represented by means of operations on formulas of the logic. It is important to realise that not every operation on formulas defines an operation on the associated meanings. An extensive discussion of this subject can be found in Janssen (1978). In 'Universal

Grammar' (Montague (1970)) a restriction is described which guarantees that the operations on logical expressions correspond with operations on meanings. Informally speaking, the new formed expression should contain the unchanged representations of the expressions operated upon. Formally in the terminology of universal algebra: the operations on logical expressions should be definable as polynomial operations over the algebra of logical expressions (usually formulated as: the operations should be polynomials). Our task can now be reformulated as follows. We have to provide for a grammar which produces step by step, and in a parallel way, both FG-structures and formulas expressing the meanings of these structures. For each basis syntactic element (i.e. lexical predicate frame), we have to give some formula, and with each syntactic rule we have to associate some (polynomially defined) operation on formulas.

3. John and every man

Let us consider the production of sentence (10)

(10) *John walks*

The production of this sentence starts with combining the basic term (11) with the verbal predicate frame (12).

(11) $(\mathrm{dlx}_i: \mathit{John}_{\mathrm{NProp}} \langle \mathrm{anim,hum,male}\rangle\ (x_i)_{\emptyset})$

(12) $\mathit{walk}_V\ (x: \mathrm{anim}(x_1))_{\mathrm{Ag}}$

According to a rule called 'term-insertion', (11) and (12) combine to (13).

(13) $\mathit{walk}_V\ (\mathrm{dlx}_i: \mathit{John}_{\mathrm{NProp}} \langle \mathrm{anim,hum,male}\rangle\ (x_i)_{\emptyset})_{\mathrm{Ag}}$

Next the syntactic function 'subject' is assigned to the term *John*, and the 'expression rules' produce sentence (10) from the thus enriched frame.

As explained in section 1, this production process has to be imitated on the level of logical formulas: for the basic frames we have to provide for formulas, for the term-insertion rule an operation on formulas. I assume that neither the assignment of syntactic functions nor the application of the expressions rules has a semantic effect. Consequently, the semantic operations corresponding to such rules are empty actions: the formulas remain unchanged. Therefore, I will neglect these two kinds of rules in the sequel.

In the logic we introduce constants for the frames (11) and (12). The constant *walk* is interpreted as a predicate of individuals, and the translation of (13) is (14), being the corresponding constant.

(14) *walk*

The constant *John* is interpreted as the element in the semantic domain which represents the individual John. One may be tempted to associate this constant with the frame for *John*. I will, however, associate a more complex formula with that frame. The reason for this has to do with the problem signalized in section 1: terms such as *John* and *every man* are to be treated on a par in the syntax, whereas the meanings of sentences (1) and (2) should differ considerably. An explanation of this approach will be given at the end of this section.

In order to be able to give the translation of the frame for *John*, predicate logic is extended with a operator λ (lamba) and with variables for predicates. The λ-operator binds variables like the quantifiers $\exists$ and $\forall$ do. The translation of (11) into the logic is (15).

(15) $\lambda P\,[P(john)]$

In order to understand this translation consider first (16), where P denotes an arbitrary predicate.

(16) $P(john)$

This formula expresses that the individual denoted by *john* has the property P. Consider now (15). By means of the symbols λP is indicated that we have to 'abstract' from the property P in the expression between the square brackets. Formula (15) denotes a function which for each property tells us whether that property holds of John or not. Let us write χ_j for the formula (15). Then χ_j is the function such that for any predicate Q:

$$\chi_j(Q) = \begin{cases} \text{true} & \text{if } Q \text{ holds for John} \\ \text{false} & \text{otherwise} \end{cases}$$

One can observe that the translation (15) of the frame for *John* is the characteristic function of his set of properties. Sloppy formulated: (15) represents the set of all Johns properties. Let us consider the value delivered by χ_j for the argument *man*.

$$\chi_j(man) = \begin{cases} \text{true} & \text{if } man \text{ holds for the argument } john \\ & \qquad \text{so if } man(John) = \text{true} \\ \text{false} & \text{otherwise, thus if } man(John) = \text{false} \end{cases}$$

From this we may conclude that

$$\lambda P\ [P(john)]\ (man) = \chi_j(man) = man(john)$$

So the value for argument α of a function expressed as $\lambda P[\beta]$ equals $\tilde{\beta}$, where $\tilde{\beta}$ is obtained from β by substituting α for each occurrence of P. This reduction is known as λ-conversion. There are certain restrictions on λ-conversion, in particular, free variables may not become bound by conversion. I will not consider such restrictions in detail.

Corresponding to the syntactic rule for term-insertion we have to provide an operation on formulas. The rule of term-insertion may have a lot of different semantic effects: term insertion in the first position of an otherwise filled predicate frame, has different semantic effect than in the second position of an otherwise empty frame. Since our approach requires for each syntactic rule a single corresponding semantic rule, the rule of term-insertion has to be split up into several rules which are syntactically closely related, but which have different semantical effects. We have as one instance of the original term insertion rule the following.

$S\text{-}TI_1$: *Term insertion 1*:

Insert a term α in a verbal predicate frame β of which only the first position is not filled.

The corresponding translation rule is:

$T\text{-}TI_1$: <u>*Translationrule for term-insertion* 1:</u>

If α',β' *are the translation of* α,β *as defined in* $S\text{-}II_1$*, then the corresponding operation on formulas yields* $(\alpha')\beta'$.

From now on we will describe such a translation rule by means of its final result, so by:

$T\text{-}SI_1$: $\alpha'(\beta')$

This translation rule says that the translation of the verb has to be the argument of the translation of the term. So the translation of (13) is as follows.

(16) $\lambda P\ [P(john)]\ (walk)$

By lambda conversion this can be reduced to (17).

(17) $walk(john)$

Frame (19) for *every man* is obtained from (18) by means of a rule called term-formation.

(18) man_N <anim,hum,male> $(x_i)_\emptyset$

(19) $(every1x_i$: man_N <anim,hum,male> $(x_i)_\emptyset$)

In the logic we have a constant corresponding with (18), which is interpretated as a predicate of individuals. Frame (18) translates into that constant, i.e. into (20).

(20) *man*

Just as was the case with term insertion, the rule for termformation is split up because several instances have their own semantic effects. Thus we have a rule

S-UT: <u>*Universal Term-formation:*</u>

Make out a noun $\alpha(x_i)$ *the term* $(every1x_i: \alpha(x_i))$

The corresponding translation rule reads:

T-UT: $\lambda P\ \forall u\ [\alpha'(u) \rightarrow P(u)]$

So the translation of (19) is

(21) $\lambda P\ \forall u\ [man(u) \rightarrow P(u)]$

This formula denotes the characteristic function of the set of properties which hold for every man. Application of S-TI_1 to (19) and (12) yields (22).

(22) $walk_V(every 1 x_i : man_N$ <anim,hum,male> $(x_i)_{\emptyset})_{Ag}$

Application to the corresponding translation rule yields (23).

(23) $\lambda P\ [\forall u\ [man(u) \rightarrow P(u)]]\ (walk)$

This reduces to (24).

(24) $\forall u\ [man(u) \rightarrow walk(u)]$

Here one observes how the first problem of section 1 can be dealt with: the two terms *John* and *every man* are syntactically treated the same, and nevertheless the different meanings (18) and (24) are obtained. This effect is due to the use of the λ-operator. It is a powerful device which makes it possible to have in the syntax an insertion on a certain position, whereas for the logical formula the effect can be obtained of a substitution on the semantically desired place (which may differ for different applications of the rule). Referring to this power Partee once said: 'Lambdas really changed my life' (lecture for the Dutch Association for Logic, Amsterdam 1980).

4. Love a woman

Using the FG rule for term formation we may produce out of (25) the term (26).

(25) $woman_N$ <anim,hum,fem> $(x_i)_\emptyset$

(26) $(i1x_i$: $woman_N$ <anim,hum,fem> $(x_i)_\emptyset)$

By inserting (26) and the frame for *John* (frame (11)) into (27) one obtains (28).

(27) $love_V(x_1$: $anim(x_1))_{Po/\emptyset}$ $(x_2)_{Go}$

(28) $love_V$(d1x: $John_{NProp}$ <anim,hum,male> $(x_i)_\emptyset)_{Po/\emptyset}$
$(i1x_j$: $woman_N$ <anim,hum,fem> $(x_j)_\emptyset)_{Go}$

A sentence obtained from frame (28) is (29).

(29) *John loves a woman.*

In order to deal semantically with (28) we introduce constants for (25) and (27) in the logic. The translation of frame (25) is the corresponding constant (30), a predicate of individuals.

(30) *woman*

The constant (31) for *love* is interpreted as a two-place predicate.

(31) *love*

The translation of frame (27) is, however, somewhat more complex. It is, roughly said, a two-place relation in which the arguments have to be filled in not simultaneously, but one after the other. Formally speaking, it is a higher order function which yields, when applied to an individual, a characteristic function of individuals. The translation of frame (27) is given in (32).

(32) $\lambda u\ \lambda v\ [love(u,v)]$

The formation of (26) has to be dealt with by means of a separate instantiation of the FG rule for term formation.

S-ET: <u>*Existential term formation*</u>

Out of the noun $\alpha(x_i)$ *make the term frame* $(ix_i$: $\alpha(x_i))$

The corresponding translation-rule is:

T-ET: $\lambda P\ [\exists u\ [\alpha'(u) \wedge P(u)]]$

So the translation of (26) is

(33) $\lambda P\ [\exists u\ [woman(u) \wedge P(u)]$

This formula is interpreted as the characteristic function of the predicates which hold for at least one woman.

By first inserting the term a *woman* into frame (27), we obtain a predicate frame of which only the first position is not filled. To such a predicate frame rule S-TI_1 can be applied. So in order to produce (28), we need a new term-insertion rule which fills the second position of (27), and a corresponding translation rule.

S-TI_2: *Term Insertion 2*
Insert term α in position 2 of verbal predicate frame β of which only the first and second position are not filled.

T-TI_2: $\lambda z\ \alpha'(\beta'(z))$

Application of S-TI_2 to (26) and (27) yields

(34) $love_V(x_1{:}anim(x_1))_{Po/\emptyset}(i1x_j{:}\ woman_N\langle anim,hum,fem\rangle\ (x_j)_\emptyset)_{Go}$

The translation of (34) is obtained by application of T-TI_2 to (33) and (32), yielding (35). This reduces by three times λ-conversion to (36).

(35) $\lambda z\ [[\lambda P\ \exists u\ [woman(u) \wedge P(u)]]\ (\lambda u\ \lambda v\ [love(u,v)](z)]$

(36) $\lambda z\ [\exists u\ [woman(u) \wedge love(z,u)]]$

Application of S-TI_1 to (34) and to the frame (11) for John yields (28). The corresponding semantic action consist in the application of T-TI_1 to (15) and (36). This yields (37).

(37) $\lambda P\ [P(john)](\lambda z\ \exists u\ [woman(u) \wedge love(z,u)])$

This reduces by twice application of λ-conversion to

(38) $\exists u\ [woman(u) \wedge love(john,u)]$

Instead of inserting into (34) the frame (11) for *John*, we might also insert frame (19) for *every man*. This yields (39).

(39) $love_V(every1x_i\text{: } man_N$ <anim,hum,male> $(x_i)_\emptyset)_{Po/\emptyset}$
$(i1x_j\text{: } woman_N$ <anim,hum,fem> $(x_j)_\emptyset)_{Go}$

Application of $T\text{-}TI_1$ to (36) and the translation of *every man* (21) yields (40),

(40) $\lambda P\ \forall u\ [man(u) \rightarrow P(u)]\ (\lambda z\ [\exists u\ [woman(u) \wedge love(z,u)]]$

which reduces by λ-conversion to (41).

(41) $\forall u\ [man(u) \rightarrow \lambda z\ [\exists u\ [woman(u) \wedge love(z,u)]](u)]$

Another application of λ-conversion to (41) would have as consequence that the rightmost u, which occurs free in the subexpression (u), would become bound by $\exists u$, as in (42).

(42) $\forall u\ [man(u) \rightarrow \exists u\ [woman(u) \wedge love(u,u)]]$

Therefore we first rename the variable bound by $\exists u$. We replace formula (41) by (43) which represents the same meaning.

(43) $\forall u\ [man(u) \rightarrow \lambda z\ [\exists v\ [woman(v) \wedge love(z,v)]](u)]$

Now we may reduce further, and after λ-reduction we obtain (44).

(44) $\forall u\ [man(u) \rightarrow \exists v\ [woman(v) \wedge love(u,v)]]$.

One can observe that the production of frame (34) for *Every man loves a woman* given above, yields the first one of the two readings (8) and (9) mentioned in section 2. In order to obtain the second reading we have to produce frame (39) in some other way as well. The first method one might think of is to perform the term insertion in the opposite order: first inserting the term *every man*, and next the term *a woman*. It is not too difficult to give such syntactic rules, and the corresponding translation rules. Extending such an approach to more complex constructions might not be that easy. On the other hand the method introduced in PTQ easily deals with the more complex constructions and has the advantage that it can be used for the treatment of several other phenomena: anaphoric pronouns, and the de-dicto de-re ambiguity. Therefore I follow Montague's method. This method is as follows.

The lexicon is extended with a new class of terms. These terms are frames consisting of a variable only, and they differ with respect to the index of the variable. Examples of such terms are (x_7) and (x_{27}). These variables can be called syntactic variables in order to discriminate them from variables in the logic; mostly the context will make clear what is intended. The syntactic variables have no counterpart in ordinary English, and in the course of the process of producing a sentence, they are either extended to a full term, or they get the status of anaphoric variables.

In the syntax the variables are treated like all terms. So in the same way as we produced (34), we can now produce (45), applying $S\text{-}TI_2$.

(45) $love_V(x_1\text{: anim}(x_1))_{Po/\emptyset}\ (x_{27})_{Go}$

The translation of a syntactic variable is a formula containing an unbound logical variable over individuals bearing the same index. Frame (46) is translated into (47).

(46) (x_{27})

(47) $\lambda P\ [P(u_{27})]$

So frame (45) translates according to $T\text{-}TI_2$ into (48).

(48) $\lambda P\ [P(u_{27})\ (\lambda u\ \lambda v\ [love(u,v)])$

By λ-conversion this reduces to

(49) $\lambda v\ [love(u_{27},v)]$

One may apply S-TI_1 to (45) and the frame (19) for *every man,* thus obtaining (50).

(50) $love_V(everylx_i: man_N$ <anim,hum,male> $(x_i)_\emptyset)_{Po/\emptyset}$ $(x_{27})_{Go}$

The corresponding translation reduces to

(51) $\forall u\ [man(u) \rightarrow love(u,u_{27})]$

Furthermore a syntactic rule is introduced which has the effect of expanding syntactic variables to full terms or to anaphoric pronouns. This rule is in fact a rule scheme which for every possible value of the involved index n constitutes a rule. The effect of the rule resembles the effect of quantification in logics, in the aspect that the rule 'binds' a variable. Therefore it is called a quantification rule.

S-Qn: <u>*Quantification rule for index* n</u>

Make a new predication out of term α *and predication* ϕ *which contains an occurrence of the term* (x_n), *by expanding the first occurrence of* (x_n) *to* α, *and expanding the other occurrences of* (x_n) *to* (Ax_n).

By 'first occurrence' I understand the leftmost (x_n) in the highest predicate frame (in section 5 this detail will be discussed). Expanding (x_n) to α has to be understood as substituting α for x_n, while replacing the index of the x in the term by n (an alternative would be to require in S-Qn that the index of the variable in the term bears index n). The anaphoric variable (Ax_n) is by the expression rules developed to an anaphoric pronoun.

Using instance S-Q17 of rule scheme S-Qn we may produce out of (50) and (26) the frame (28). Thus we produced (20) along two different production processes.

The translation rule (or rather scheme) corresponding with S-Qn is:

T-Qn: $\alpha'(\lambda u_n \phi')$

The trânslation of (28) - if produced in the way just indicated - is obtained by combining (33) and (51) in accordance with T-Q17, yielding (52).

(52) $\lambda P\ [\exists u\ woman(u) \wedge P(u)]\ (\lambda u_{27}\ \forall u\ [man(u) \rightarrow love(u,u_{27})])$

After renaming bound variables, this reduces to (53).

(53) $\exists u\ [woman(u) \wedge \forall v\ [man(v) \rightarrow love(u,v)]]$

This represents the desired second reading of (28). The two production processes lead to the same frame, but resulted in different translations representing (in this case) two different meanings.

Another phenomenon for which S-Qn can be applied is coreferentiality. Let us first produce (54) and (55).

(54) $walk_V\ (x_{17})_{Ag}$
(55) $talk_V\ (x_{17})_{Ag}$

Their respective translations are (56) and (57).

(56) $walk(u_{17})$
(57) $talk(u_{17})$

Let us then combine these frames by the rule for conjunction.

S-C: _Conjunction rule_
Make out of predication α and predication β the predication α and β.

The corresponding translation rule reads:

T-C: $\alpha' \wedge \beta'$

Application of S-C to (54) and (55) yields (58).

(58) *walk* $(x_{17})_{Ag}$ *and talk* $(x_{17})_{Ag}$

Next we combine (58) with frame (11) for *John* according to S-Q17, producing a frame from which we can obtain sentence (59).

(59) *John walks and he talks*

Application of T-C to (56) and (57) yields

(60) $walk\ (u_{17}) \wedge talk\ (u_{17})$

Combining this with the translation (15) of *John* according to T-Q17 yields

(61) $\lambda P\ [P(john)]\ (\lambda u_{17}\ [walk(u_{17}) \wedge talk(u_{17})])$

reducing to

(62) $walk(john) \wedge talk(john)$

Below I will give some translations of frames and rules which indicate how parts of Dik (1980) I have not yet considered, can be dealt with. I will not, however, present rules for phenomena which either require a complex semantic treatment (such as 'question formation' - see Groenendijk and Stokhof (1980), 'numbers' - see Bartsch (1973), and 'tense' - see Janssen (1980b)), or which have not yet been dealt with adequately in Montague grammar (such as 'satellites' - see section 5). The translations for the frames for the non-anaphoric pronoun *he*, for *give*, and for *believe* are given in (62), (63) and (64) respectively (since *believe* is an intensional verb, a fully correct translation should contain an intensional operator).

(61)	$\lambda P\ [P(c_2)]$	c is a constant of the type of individuals
(62)	$\lambda u\ \lambda v\ \lambda w\ [give(u,v,w)]$	
(63)	$\lambda u\ \lambda P\ believe(u,P)$	P is a variable of the type of translations of sentences.

Finally I will just mention some rules.

S-DT: <u>*definite term formation:*</u>

Make out of the noun frame $\alpha(x_i)$ *the term frame* $(\mathrm{dl}x_i\colon \alpha(x_i))$

T-DT: $\lambda P\ \exists u\ [\forall v\ [\alpha'(v) \leftrightarrow u = v] \wedge P(u)]$

S-RC: *Relative clause formation:*

Make a noun-frame of the form $\alpha(x_i):\phi$ *as follows. Combine the noun frame* $\alpha(x_i)$ *with the predication* ϕ *(which has to contain the syntactic variable* x_n*) by expanding* x_n *to a suitable relative pronoun.*

T-RCn: $\lambda u_n\,[\alpha'(u_n) \wedge \phi']$

S-TI_3: *Term Insertion 3:*

Insert term α *in position 3 of a verbal predicate frame in which positions 1,2,3 are empty.*

T-TI_3: $\lambda z_1\,\lambda z_2\,[\alpha(\beta(z_1)(z_2))]$

5. Discussion

In this section I will make some general remarks about the relation between functional grammar and Montague grammar. The first remark I want to make, concerns the difference between the version of functional grammar as it figures in the previous sections and its more standard form. I did not make use of all the information present in a predicate frame. Semantic functions (e.g. Goal), features (e.g. human), and variables (the x_n's in frames) were not taken into consideration.

Semantic functions express semantical details, for example that a certain position of a frame will contain the goal, and another one the locative satellite. The kind of semantics currently used in Montague grammar is not yet that refined enabling to take these differences into account. In order to do so the logic and the model probably would have to be enriched. We had a limited aim (see section 1), for which it was not necessary to bother about these details. For larger fragments of English even reaching this limited aim is even difficult enough.

The features provide the information what kind of terms can be put into which positions. Certainly, Montague grammar will have to deal with such selection restrictions as well, and the information contained in these features has to be incorporated somehow. It has both been proposed to do so in the semantics by means of partially defined functions (Cooper (1975), Waldo (1979)), and to to so in syntax by means of subcategorization (Janssen (1980b)). Either one of these methods could be followed here.

The role of the variables in functional grammar is taken over, in a new way, by the quantification rules. Since functional grammar was not designed together with an explicit formal semantics, it is not too surprising that some details had to be changed.

A parallel between functional grammar and Montague grammar is the tendency to eliminate transformations. In transformational grammar it was often their semantic relation that motivated a transformational relation between sentences. The basic difference between Montague grammar and transformational grammar is that Montague grammar has an explicit semantic component. Therefore Montague grammar has the possibility to formalize such relations among sentences at the level of semantics. Sentences may be produced independently syntactically, while their semantic relations can still be dealt with, but now in the semantic component. Thomason (1976) treats passive sentences as produced independently of the active sentences, and accounts for the relation between them in semantics. Gazdar (1980) propagates a context-free - i.e. a non-transformational - approach to syntax. Many of his rules are possible because his grammar has an explicit semantic component analogous to that of Montague grammar. Several governed transformations are treated non-transformationally in Dowty (1979), and Bartsch (1979), again by having the semantics do the job. So there is a general tendency among Montague grammarians to avoid transformations. Not using transformations is one of the aims of functional grammar. It is to be expected that many of the results in the field of Montague grammar can be applied in functional grammar as well. The following is an example.

How could one allow for the possibility that both (62) and (63) are produced?

(62) *John serves the cake to Mary.*

(63) *John serves Mary.*

In transformational grammars (63) usually is obtained from a source like (62) by means of a transformation called 'object deletion'. Dowty (1979) does not use a deletion rule. Instead he proposes a rule which makes out of the three place verb *serve,* a two place verb. This rule has no visible syntactic effect: the verb obtains the status of a two-place verb without changing its form. Semantically, this rule has the effect of introducing some existential quantifier. The same idea can be applied in functional grammar: deleting nothing, just change the status of the verbal frame (e.g.

by inserting a dummy term or removing the third position).

An important difference between functional grammar and Montague grammars as they are usually described the literature is that, whereas a Montague grammar produces rather natural phrases, a functional grammar produces abstract structures in the non-final stages. This has as consequence that in a Montague grammar two only superficially different phrases such as (64) and (65) are to be considered as different syntactic objects.

(64) *give Mary a book*

(65) *give a book to Mary*

In the more abstract approach of FG they correspond to the same predication, leaving it to the expression rules to make the difference. Another example of this phenomenon is the active/passive distinction. A functional grammar can have somewhat simpler rules since superficial differences have not to be accounted for in the stage where the meaning is determined.

An advantage of the abstract structures becomes apparent if we consider syntactic variables. A Montague grammar produces strings of words, so a variable has to be represented as a word. They are represented as a male pronoun to which an index is attached: by he_n and by him_n. This representation has the disadvantage that an arbitrary choice is made: the variables look 'male' and 'singular'. By later rules these decisions often have to be withdrawn. Furthermore one is tempted to consider the variables syntactically as pronouns, which they are not (Janssen (1980a)). Functional grammar has the advantage that syntactic variables can be represented as what they really are: abstract elements. No premature decisions have to be taken.

These syntactic advantages have their price. The frames of a functional grammar are rather distinct from real sentences. One might ask whether the available abstract information is the kind of information one needs for producing correct sentences. In any case, this is not the kind of information which traditionally is considered to be of the required kind: a constituent structure. Kwee Tjoe Liong has developed a computer program implementing Dik (1981). Consequently he had to be very explicit, in particular in the formulation of the expression rules. There turned out (Kwee (1979) and Kwee (this volume)), to be a lot of unsolved problems in formulating rules producing sentences out of frames. The fact that the available structural information is rather abstract, brought me into problems when formulating of S-Qn. I will turn to these problems now.

Pronominal reference is always possible from right to left: a suitable pronoun can always refer to a term occurring earlier in the sentence. In certain circumstances pronominal reference also is possible in the other direction. But it is rather difficult to characterize these situations. So in case there are two positions in a sentence, one to be filled with a term, and the other with a pronoun referring to that term, then a safe strategy is to put the term in the leftmost position. This safe strategy is followed in PTQ in the formulation of the rule corresponding to S-Qn. When formulating S-Qn in FG this strategy cannot be applied since it is not possible to say in advance in which linear sequence the terms will occur in the sentence which is finally obtained. The assignment of syntactic functions may have as consequence that anaphoric variables are raised from deep embeddings and the expression rules can make that the corresponding anaphoric pronouns occur earlier in the sentence than the terms they refer to. I do not know how to formulate S-Qn in such a way that only correct reference patterns result. It is not attractive to try to solve this problem by allowing 'unrestricted' expansions in S-Qn, and to put restrictions on the rules assigning syntactic functions and the expression rules, which would require that pronouns should occur to the right of the term they refer to. For this would introduce a filter into the grammar since there is no guarantee that from a given frame with several pronouns a sentence can be produced which obeys the requirements. Such a filter is undesirable from the point of view of functional grammar itself. (See section 2.1 in Dik (1981)).

The safe strategy followed in Montague grammar does not guarantee that all the sentences which are produced are correct. Although a pronoun may always refer backwards, sometimes it has to be a personal pronoun, and sometimes a reflexive pronoun. Since the rules of PTQ do not treat reflexives, the sentence *John sees him* is produced, with the meaning that John sees himself. It is difficult to characterize the positions in which a reflexive is required. It is even more difficult to characterize the positions where forward referring is allowed. In transformational grammars there has been some progress in characterizing these two kinds of positions. These characterisations use structural properties (Reinhart (1979)), such as the notion C-command. In a PTQ-style grammar plain strings are produced without any structure; here a correct treatment of pronouns seems impossible. But one might enrich these strings with markers indicating the relevant structural information (Bennett (1976)). It would be more straightforward to have the grammar produce trees (or equivalently labelled bracketings) instead of plain strings. If these trees are of the same as the trees used in transformational grammars, then all insights from that field can be used in Montague grammar as well. The idea of such a grammar is due to Partee (1973), and has been worked out in Partee (1979a,b) and Bach (1979).

In a functional grammar structures are produced of a completely different kind, so the insights from transformational grammars cannot easily be adopted. This leaves the challenge for the functional grammarians to characterize in the terminology of functional grammar the configurations where reflexives or forward referring pronouns may occur. This seems to me a very difficult task, since a frame does not give much information about the surface forms it can take.

Let me summarize my views on the relation between functional grammar and Montague grammar. There is no problem in considering functional grammar to be an instance of the framework of Universal Grammar (Montague (1970)). This has the consequence that meanings are not considered as to be determined by frames as such, but only given by their derivational histories. Derivational histories as they are designed in Montague grammar, can be imitated in functional grammar. Sometimes a simplification is possible because of the rather abstract structures produced in a functional grammar. This same fact, however, leaves certain syntactic problems to be solved. Of course, such an imitation may have the effect that certain production processes of functional grammar have to be changed; an example is the relative clause construction given in section 4 (for a discussion see Partee (1973) and Janssen (1980a)). In any case the program for dealing with semantics in a functional grammar is established. One has to provide for formulas for the basic frames, and for each rule a (polynomially defined) operation on formulas.

Acknowledgements

I wish to thank Simon Dik, Kwee Tjoe Liong, and especially Jeroen Groenendijk for their comments on a previous version of this paper.

REFERENCES

BACH, E.
1979 Montague grammar and classical transformational grammar. In Davis & Mithun 1979, pp.3-49.

BARTSCH, R.,
1973 The semantics and syntax of number and numbers. In J.P. Kimball (ed.) *Syntax and semantics. Volume 2.* Academic Press, New York, 1973, pp. 51-93.

1978 Infinitives and the control problem in categorial grammar. In *Theoretical Linguistics* 5, 217-250.

BENNETT, M.
1976 A variation and extension of a Montague fragment of English. In Partee 1976, pp. 119-163.

COOPER, R.H.
1975 *Montague's semantic theory and transformational syntax.* Dissertation, Univ. of Massachusetts, Amherst. Published by: Xerox University Microfilms.

DAVIS, S. & M. MITHUN (eds.)
1979 *Linguistics, philosophy and Montague grammar. (Proc. conf. Albany 1977).* Univ. of Texas Press, Austin, Texas.

DIK, S.C.
1981 Seventeen Sentences: basic principles and application of functional grammar. In *Proc. Conf. on current approaches to syntax, Milwaukee (Wisc.) 1979*, Syntax and Semantics 13, Academic Press, to appear.

DOWTY, D.
1978 Governed transformations as lexical rules in a Montague grammar. In *Linguistic Inquiry* 9, 393-426.

1979 Dative movement and Thomason's extensions of Montague grammar. In Davis & Mithun 1979, pp. 153-222.

GAZDAR, G.
19xx Phrase structure grammar. In G.K. Pullum & P. Jacobson (eds.) *On the nature of syntactic representation,* to appear.

GROENENDIJK, J. & M. STOKHOF
1978 *Proc. of the second Amsterdam coll. on Montague grammar and related topics.* Amsterdam papers in formal grammar II, Centrale Interfaculteit, Univ. of Amsterdam.

1980 The semantics of wh-complements. In Groenendijk, Janssen, Stokhof 1980.

GROENENDIJK G., T. JANSSEN & M. STOKHOF (eds.)
1980 *Formal methods in the study of language. Proceedings of the third Amsterdam colloquium.* Mathematical Centre Tract, Mathematisch Centrum, Amsterdam.

GUENTHER, F. & J.S. SCHMIDT (eds.)
1979 *Formal semantics and pragmatics for natural languages.* Synthese language library 4, Reidel, Dordrecht.

HENY, F. & H.S. SCHNELLE (eds.)
1979 *Selections from the third Groningen round table,* Syntax and Semantics 10, Academic Press, New York.

JANSSEN, T.M.V.
1978 Compositionality and the form of the rules in Montague grammar. In Groenendijk & Stokhof 1978, pp. 101-124.

1980a Compositional semantics and relative clause formation in Montague grammar. In Groenendijk, Janssen, Stokhof 1980.

1980b On problems concerning the quantification rules in Montague grammar. In Rohrer 1980, pp. 113-134.

KWEE Tjoe Liong,
1979 *A68-FG(3). Simon Dik's funktionele grammatika geschreven in ALGOL68, Versie nr. 3* Publicatie 22, Instituut voor algemene Taalwetenschap, University of Amsterdam.

(this volume) In search of an appropriate relative clause.

LEWIS, D.
1972 General semantics. In D. Davidson & G. Harman (eds.) *Semantics of natural language*, Synthese library 40, Reidel, Dordrecht, 1972, pp. 169-248. Reprinted in: Partee 1976, pp

MONTAGUE, R.
1970 Universal grammar, *Theoria* 36, 373-398. Reprinted in Thomason 1974, pp. 222-246.

1973 The proper treatment of quantification in ordinary English. In K.J.J. Hintikka, J.M.E. Moravcsik & P. Suppes (eds.) *Approaches to natural language*. Synthese Library 49, Reidel, Dordrecht, 1973, pp. 221-242.

PARTEE, B.H.
1973 Some transformational extensions of Montague grammar. *Journal of Philosophical Logic* 2, 509-534. Reprinted in: Partee, 1976, pp.51-76.

1976 *Montague grammar*. Academic Press, New York.

1979a Constraining transformational Montague grammar: A framework and a fragment. In Davis & Mithun 1979, pp. 51-102.

1979b Montague grammar and the well-formedness constraint. In Heny & Schnelle 1979, pp. 275-314.

REINHART, T.
1979 Syntactic domains for semantic rules. In F. Guenther & J.S. Schmidt 1979, pp. 107-130.

ROHRER, C.
1980 *Time, tense and quantifiers. (Proc. conf. Stuttgart 1979)*. Linguistische Arbeiten 83, Max Niemeyer Verlag, Tuebingen.

THOMASON, R.H.
1974 *Formal philosophy. Selected papers of Richard Montague*. Yale University Press, New Haven.

1976 Some extensions of Montague grammar. In Partee 1976, pp. 77-117.

WALDO, J.
1979 A PTQ-semantics for sortal incorrectness. In Davis & Mithun 1979, pp. 311-331.