

MONTAGUE AND CATEGORIAL GRAMMAR

Helmut Frosch

Montague grammar (MG) designates a field of formal linguistics which is based on the works of the logician and philosopher Richard Montague from the early 1970s. MG takes up the tradition of mathematical linguistics established by Chomsky and others in referring to both natural and artificial languages as objects to be described with the help of mathematical methods. MG, however, goes beyond these traditional approaches by formalizing even the semantic and pragmatic aspects of these languages as rigorously as had been the case before only with syntax.

1. Syntax of MG

A simple fragment of English (EF) may include sentences such as (1):

(1) *necessarily John talks*

This sentence can be analyzed syntactically with the help of structuralist methods and can thus be broken down into its constituent structure. The bracketing shows such constituents:

(2) [*necessarily [John talks]*].

What was new about Chomsky's generative grammar was that it did not only describe such structures but provided a formal grammar generating these structures. MG also uses mathematical methods in syntax, but the difference with Chomsky's – technically speaking – formal grammars is that MG reconstructs syntax algebraically. Structures belonging to a language L are called 'proper expressions' of L in MG. They are made up of (mostly shorter) expressions by so-called 'structural operations'. The proper expressions of a language generated by such structural operations F_ν may be fairly complicated entities such as graphic patterns, if for example the language under consideration is a graphical user interface of a computer. In the case of EF the following relatively simple operations of F_1 and F_2 apply:

(OP1) $F_1(\alpha, \beta) = [\beta \alpha']_1$

where α' comes from α by:

- (i) if β is a term, and α contains at least one verb, the first verb in α is replaced by that form of the verb which agrees with β in person and number;
- (ii) otherwise $\alpha' = \alpha$.

(OP2) $F_2(\alpha, \beta) = [\alpha \beta]_2$

With the help of these operations it is possible to form (2):

$F_1(\textit{talk}, \textit{John}) = [\textit{John talks}]_1$

$F_2(\textit{necessarily}, [\textit{John talks}]_1) = [\textit{necessarily [John talks]}]_2$

Operation F_2 simply concatenates its arguments and puts brackets around the resulting expressions. The brackets are indexed with the corresponding operation number. This operation largely corresponds to a phrase structure rule within generative grammar. Operation F_1

concatenates its arguments in reverse order, puts indexed brackets around the result and establishes the agreement between subject and verb. Although this process has not been explicitly defined above, it becomes clear that operation F_1 is a combination of a phrase structure rule and transformational rules.

MG requires that the generated language L be a disambiguated language. This implies above all that there must be only one way to generate every proper expression of L with the help of the operations F_γ . This is why the brackets in the proper expressions of EF are indexed with the numbers of operations which generate them. If this were different, example (2) might be generated via

$$F_1(\text{talk}, \text{John}) = [\text{John talks}]$$

$$F_2(\text{necessarily}, [\text{John talks}]) = [\text{necessarily } [\text{John talks}]]$$

or, for example, via

$$F_1(\text{talk}, \text{John}) = [\text{John talks}]$$

$$F_1([\text{John talks}], \text{necessarily}) = [\text{necessarily } [\text{John talks}]]$$

without the resulting expression indicating its origin. In other words, there would be no unique reading for the expression under consideration.

Besides structural operations two further components belong to the syntax of a MG: first a relation R which relates the proper expressions of a disambiguated language L to their syntactically ambiguous 'surface structures'. In the following example the relation R simply deletes the brackets, resulting in the original sentence (1):

$$[\text{necessarily } [\text{John talks }]_1]_2 \text{ R necessarily John talks}$$

Secondly, the expressions of L have to be determined categorially: expressions which syntactically (and semantically) play the same 'role', are put together into a set P_δ which forms a syntactic category indexed with the corresponding category index δ . Every category P_δ contains a set B_δ of basic expressions belonging to this category as a subset.

The language EF , from which sentence (1) is an example contains the following sets of basic expressions:

$$B_T = \{\text{John}, \text{Mary}, \text{I}, \text{you}\}$$

$$B_{IV} = \{\text{talk}\}$$

$$B_{TV} = \{\text{see}, \text{love}\}$$

$$B_{AS} = \{\text{necessarily}\}$$

$$B_t = \Lambda$$

The category indices are T for 'term' (traditionally NP), IV for 'intransitive verb phrase', TV for 'transitive verb phrase', AS for 'adsentential phrase' (traditionally sentence adverbial) and t for 'truth value expression', i.e. declarative sentences. As EF does not contain declarative sentences as basic expressions, B_t is the empty set.

In addition to the basic expressions, each category P_δ contains certain expressions generated by the structural operations. These expressions are related to P_δ by so-called 'syntactic rules'. Every syntactic rule is a sequence, consisting of a syntactic operation, the category indices of the arguments of this operation, and a category index which indicates the category of the value of the operation. For EF this is:

- (S1) < F₁, IV, T, t >
 (S2) < F₂, AS, t, t >
 (S3) < F₂, TV, T, IV >

As a result, an expression generated with the help of F₁ is classified as a declarative sentence (element of P_I), if and only if the first argument of F₁ consists of an intransitive verb phrase (element of P_{IV}) and the second argument consists of a term (element of P_T). F₁(*talk, necessarily*) = [*necessarily talk*]₁, however, would be a proper expression of EF, but would not belong to any category, because *necessarily* is not an element of P_I. According to the terminology of MG the expression [*necessarily talk*]₁ is not meaningful and therefore 'filtered out'. It is important to note that both the second and the third syntactic rules refer to F₂; thus, S3 refers to the concatenation of a transitive verb phrase with a term as an object. The result is an intransitive verb phrase. Thus the result of F₂(*love, you*) = [*love you*]₂ is an intransitive verb phrase. Note also that the third person singular of *love* is generated only when a declarative sentence is formed with the help of F₁([*love you*]₂, *Mary*) = [*Mary [loves you]*]₂.

2. Semantics and pragmatics of MG

In MG the semantic interpretation of a language L is done either directly or indirectly. L is interpreted indirectly when it is first translated into a second language L' which is then interpreted directly. In this case, the interpretation of an expression α in L is guaranteed indirectly via the interpretation of the translation of α into the expression α' belonging to L'. Usually a language of formal logic is used as L', because the relatively simple syntactic structure in such a language provides a clear explanation of the relevant relationships of meaning. As EF is very simple in itself, this language fragment is interpreted directly.

Such an interpretation requires a model **M** which determines the denotations of meaningful expressions of EF. In particular, **M** determines whether a declarative sentence is true in **M**: **M** thus represents the world beyond language which is being talked about in the language EF. Although the language EF is syntactically rather simple, it contains the modal expression *necessarily* and the indexicals *I* and *you* which require a nontrivial model **M** for EF. In order to indicate whether a declarative sentence containing *necessarily* is true, it is not enough to look at the world as it is. Rather, every possible (situation of the) world must be taken into consideration. This is why **M** contains a set of possible worlds. In order to say whether an utterance of a declarative sentence containing *I* is true, it is not enough to give information about the situation of the world. It also has to be clear who the speaker is in the current context. Therefore **M** also contains a set of possible contexts of use.

The model **M** consists of the components A, I, J and INT^{<ij>} with A being a nonempty set, i.e. the domain of individuals of the model. I is a nonempty set of possible worlds, J is a nonempty set of possible contexts of use. For every point of reference <i, j> (an ordered pair consisting of a world i and a context j) INT^{<ij>} is a function which interpretes the meaningful expressions in EF, i.e. which assigns denotations to them.

First, certain basic expressions of EF are interpreted as follows:

- (IN1) If $\alpha \in B_T$ (i.e. α is an element of B_T), then:
 (i) if $\alpha = \textit{John}$, or $\alpha = \textit{Mary}$, then $\text{INT}^{\langle ij \rangle}(\alpha) \in A$, and for all $i' \in I$ and $j' \in J$:
 $\text{INT}^{\langle i'j' \rangle}(\alpha) = \text{INT}^{\langle ij \rangle}(\alpha)$;
 (ii) $\text{INT}^{\langle ij \rangle}(I)$ is the speaker in j, and $\text{INT}^{\langle ij \rangle}(\textit{you})$ is the addressee in j, such that for all $i' \in I$:
 $\text{INT}^{\langle i'j' \rangle}(I) = \text{INT}^{\langle ij \rangle}(I)$, and $\text{INT}^{\langle i'j' \rangle}(\textit{you}) = \text{INT}^{\langle ij \rangle}(\textit{you})$.

- (IN2) If $\alpha \in B_{IV}$, then $INT^{<ij>}(\alpha)$ is a function that assigns to each element of A one of the truth values 0 ('false') or 1 ('true'), such that for all $j' \in J$:
- $$INT^{<ij>}(\alpha) = INT^{<ij'>}(\alpha).$$
- (IN3) If $\alpha \in B_{TV}$, then $INT^{<ij>}(\alpha)$ is a function that assigns to each element of A a function from elements of A to truth values, such that for all $j' \in J$:
- $$INT^{<ij'>}(\alpha) = INT^{<ij>}(\alpha).$$

Regardless of the contexts and possible worlds, proper names are assigned to the individuals which within the model M bear these names. They are therefore 'rigid designators' in the sense of Kripke. Depending on the context, personal pronouns are assigned different individuals which, however, are the same in all worlds. Intransitive and transitive verbs roughly denote one-place or two-place relations (or, equivalently, in this case characteristic functions and functions to characteristic functions) between individuals. As these can change with the situation of the world (but not with the context) they are assigned different denotations for different worlds.

Second, the interpretation of all complex expressions of EF is defined by:

- (IN4) If $\alpha \in P_{IV}$ and $\beta \in P_T$, then:
- $$INT^{<ij>}([\beta \alpha']_1) = INT^{<ij>}(\alpha)(INT^{<ij>}(\beta)), \text{ where } \alpha' \text{ is as in (OP1).}$$
- (IN5) If $\alpha \in P_{TV}$ and $\beta \in P_T$, then:
- $$INT^{<ij>}([\alpha \beta]_2) = INT^{<ij>}(\alpha)(INT^{<ij>}(\beta)).$$
- (IN6) If $\alpha \in P_v$, then $INT^{<ij>}([\textit{necessarily } \alpha]_2) = 1$ iff (if and only if) $INT^{<i'j>}(\alpha) = 1$ for all $i' \in I$.
- (IN7) An utterance of a declarative sentence α is true at the point of reference $<ij>$ (with respect to the model M) iff $INT^{<ij>}(\alpha) = 1$. A declarative sentence α is true in the world i (with respect to the model M) iff $INT^{<ij>}(\alpha) = 1$ for all $j \in J$.

These definitions show that the semantics of MG is compositional, i.e. the interpretation of complex expressions is derived functionally from the interpretation of the partial expressions contained in them. This means in particular that the context-dependent interpretation of a pronoun or the 'rigid' interpretation of a name is 'passed on' to the interpretation of a declarative sentence which contains such expressions. (IN7) shows that it is also possible from a formal point of view to differentiate between whether the utterance of a sentence or a sentence *tout court* is true, if the context of use, i.e. partly pragmatic aspects, is taken into consideration in the formal interpretation. For example:

[*Mary [loves you]₂*]₁ is a declarative sentence of EF. (by the syntax of EF)

An utterance of [*Mary [loves you]₂*]₁ is true at the point of reference $<ij>$ iff

$INT^{<ij>}([\textit{Mary [loves you]}_2]_1) = 1.$ (by IN7)

$INT^{<ij>}([\textit{Mary [loves you]}_2]_1) = 1$ iff
 $INT^{<ij>}([\textit{love you]}_2)(INT^{<ij>}(\textit{Mary})) = 1.$ (by IN4)

$INT^{<ij>}([\textit{love you]}_2) = INT^{<ij>}(\textit{love})INT^{<ij>}(\textit{you}).$ (by IN5)

$\text{INT}^{<i,j>}(\textit{love})$ is a function which assigns to each element of A a function from elements of A to truth values, such that for all $j' \in J$:

$$\text{INT}^{<i,j'>}(\textit{love}) = \text{INT}^{<i,j>}(\textit{love}). \quad (\text{by IN3})$$

$\text{INT}^{<i,j>}(\textit{you})$ is the addressee in j , such that for all $i' \in I$:

$$\text{INT}^{<i',j>}(\textit{you}) = \text{INT}^{<i,j>}(\textit{you}). \quad (\text{by IN1})$$

$\text{INT}^{<i,j>}(\textit{Mary})$ is Mary, and for all $i' \in I$ and $j' \in J$:

$$\text{INT}^{<i',j'>}(\textit{Mary}) = \text{INT}^{<i,j>}(\textit{Mary}). \quad (\text{by IN1})$$

As *you* denotes the addressee in context j , the interpretation of $[\textit{love you}]_2$ also depends on the context: it is a function which assigns the value 1 to all individuals who love the individual addressed in the context j . If Mary is one of these individuals, an utterance of the sentence $[\textit{Mary} [\textit{loves you}]_2]_1$ is true at the point of reference $\langle i, j \rangle$. At the same time the sentence $[\textit{Mary} [\textit{loves you}]_2]_1$ may be wrong in a world i , if there is a context j' in which the addressee in j' is not loved by Mary. The sentence $[\textit{Mary} [\textit{loves John}]_2]_1$, however, which does not contain context-dependent expressions, is true in a world i , if there is at least one utterance of it which is true at a point of reference $\langle i, j \rangle$. Such a sentence may even be true if it has never been uttered. It is sufficient that its truth value is 1 in at least one context, for then its truth value is 1 in all contexts. Whether it actually makes sense to assume that there are sentences which are never uttered, is a totally different question which is by no means predetermined by (INF7).

3. Categorical grammar

The term 'categorical grammar' (CG) is a summary expression for various interrelated formalisms for the description of natural languages. Such formalisms had already been developed during the thirties by the Polish logician Ajdukiewicz and others. As opposed to MG and generative grammar, the initial idea was not to develop a formal system for the generation of sentences but to create a formalism which determines whether a sequence of words is a sentence or not. However, today CG formalisms are used to generate sentences as are other grammatical systems.

What probably led to the development of CG was the fact that specific expressions in a language (words and word sequences) can only be combined with a limited number of other expressions. This can be explained by the assumption that in natural languages there are functor argument structures, as for example in predicate logic predicates are functors taking arguments. In sentence (1)

(1) *necessarily John talks*

necessarily may be considered a functor which takes the sentence *John talks* as its argument. *talks* is a functor with the name *John* as its argument. Expressions which play the same role as functors or arguments can be summarized in so-called 'categories'. In this case, the categories are sentences, names and intransitive verb phrases.

A CG is developed by assigning symbols to the categories which by their mere form determine the role of an expression within a certain category. Names and sentences only occur as arguments; they are called 'basic categories'. For basic categories simple category symbols are used, such as N for names and S for sentences. Categories of expressions occurring as functors are called 'derived categories'. Category symbols for derived categories are combined symbols, e.g. S/N for intransitive verb phrases. The crucial point is to see that the symbol by itself emphasizes that an intransitive verb phrase takes a name (more generally: a noun phrase) as its

argument and that the resulting expression makes up a sentence. Correspondingly, the category of sentence adverbials has the symbol S/S, because in connection with a sentence any sentence adverbial makes up another sentence. This is the so-called 'cancellation principle' of CG.

Based on the simple symbols of the basic categories, the symbols for any derived categories can be created with the help of the following recursive definition:

(CS) S and N are category symbols.

If σ and τ are any category symbols, then (σ/τ) is a category symbol.

Nothing else is a category symbol.

Since according to (CS) S and N are category symbols, (S/N), (S/S), ((S/N)/N), ((S/S)/((S/N)/N)) etc. are also category symbols according to (C/S). The outer brackets are usually left out in order to ensure better readability. For example we write S/N instead of (S/N).

A category symbol such as (S/N)/N indicates that in combination with an expression of category N (*John*) an expression of this category (*loves*) makes up an expression of the category S/N (*loves John*). In combination with a second name (*Mary*) an expression of the category S/N (*loves John*) makes up an expression of category S (*Mary loves John*). As the order of combination differs in both cases, this order must be determined. The usual method of doing this is to introduce a bidirectional category system whose category symbols by their form also explain whether the argument in question is to be connected to the expression on the left- or on the right-hand side. Category symbols of the form (σ/τ) indicate that expressions of a category τ connect to expressions of a category (σ/τ) on the right-hand side, whereas $(\tau\backslash\sigma)$ is for the left-hand side. In correspondence with the SVO word order in English, transitive verbs of this language belong for example to category (N\S)/N.

In formal terms, every CG thus consists of two parts: first the categories and their symbols, and second a 'mechanism' which creates the expressions contained in a language. Such a mechanism is usually a context-free phrase structure grammar which, as in the example mentioned, has the following rule schemata

$$\begin{aligned}\sigma &\rightarrow (\sigma/\tau) \tau \\ \sigma &\rightarrow \tau (\tau\backslash\sigma)\end{aligned}$$

providing a characterization of all production rules of the grammar.

However, every CG may also be considered a special case of MG. For the fragment of English EF given above it is sufficient to write the category indices T, TV, IV and t as N, (S/N)/N, S/N and S in the terms of CG with an undirected category system as a basic assumption. All other parts of the grammar can be adopted without any changes. This leads to the following form for the syntactic rules:

$$\begin{aligned}(S1') &< F1, S/N, N, S > \\ (S2') &< F2, S/S, S, S > \\ (S3') &< F2, (S/N)/N, N, S/N >\end{aligned}$$

Obviously, (S1') - (S3') are instances of the CG cancellation principle, i.e. they could be replaced by the principle. In general, however, this is not possible as MG also admits syntactic rules which do not obey this principle. In addition, the structural operations of MG can be far more complex than the simple concatenation of expressions, as is already clear from F1 above.

References: K. AJDUKIEWICZ (1967) Syntactic connection. In S. McCall (ed.) *Polish logic*: 207-231. Oxford Univ. Press. Y. BAR-HILLEL, C. GAIFMAN & E. SHAMIR (1960) On categorial and phrase-structure grammars.

Bulletin of the Research Council of Israel 9F: 1-16. M.J. CRESSWELL (1973) *Logics and languages*. Methuen. D.R. DOWTY, R.E. WALL & S. PETERS (1981) *Introduction to Montague semantics*. Reidel. H. FROSCHE (1993) *Montague-Grammatik*. In J. Jacobs et al. (eds.): 413-129. G. GAZDAR, E. KLEIN, G.K. PULLUM & I. SAG (1985) *Generalized phrase-structure grammar*. Blackwell. J. JACOBS, A. von STECHOW, W. STERNEFELD & T. VENNEMAN (eds.) *Syntax*. de Gruyter. S. KRIPKE (1980) *Naming and necessity*. Blackwell. D. LEWIS (1970) *General semantics*. *Synthese* 22: 18-67. G. LINK (1979) *Montague-Grammatik*. Fink. S. LÖBNER (1976) *Einführung in die Montague-Grammatik*. Scriptor. R. MONTAGUE (1970a) *English as a formal language*. In B. Visentini et al. (eds.) *Linguaggi nella società e nella tecnica*: 189-224. Ed. di Comunità. ----- (1970b) *Universal grammar*. *Theoria* 36: 373-398. ----- (1973) *The proper treatment of quantification in ordinary English*. In J. Hintikka, J. Moravcsik & P. Pappas (eds.) *Approaches to natural language*: 221-242. Reidel. ----- (1974) *Formal philosophy*. Yale Univ. Press. B.H. PARTEE (ed.) (1976) *Montague grammar*. Academic Press. M. STEEDMAN (1993) *Categorial grammar*. In J. Jacobs et al. (eds.): 395-413.

[See also: Formal pragmatics; Logical semantics; Model-theoretic semantics]