MONTAGUE GRAMMAR, CATEGORIES AND TYPES: A PRESENTATION OF ACTUAL THEORIES IN SEMANTICS AND DISCOURSE INTERPRETATION.*

Francisco J. Salguero Lamillar (University of Seville)

E-mail: salguero@us.es

1. Just a little bit of recent history about categories

1.1 Richard's Montague semiotic program

During the 60's and the 70's, one of the great questions in the linguistic discussion was the problem of how to establish the semantic component of grammar. In this context it is not strange that some linguists turned their looks toward those theories known as logical or categorial grammars, in which the formal semantic component was well defined. But, despite of the importance of the role played by the logical grammars in the "semantic controversies" during the decade 1955-1965, they didn't influence substantially in the later developments, fundamentally in those that were carried out in the generative frame. It is in this sense in which it is necessary to interpret Gerald Gazdar's and Geoffrey Pullum's following statement:

> "Categorial grammars [...] have always had a somewhat marginal status in linguistics. There has always been someone ready to champion them, but never enough people actually using them to turn them into a paradigm. The currency they have [...] is due in large measure to Montague, who based his semantic work on a modified categorial grammar" (Gazdar & Pullum 1985).

This text implies some interesting considerations on the existent relationship among Montague Grammar (MG) and Categorial Grammars (CG). In the first place, Categorial Grammars are considered as "marginal" grammatical formalisms, for what their linguistic interest is scarce and not very significant, until the point of not having been able to never become a grammatical paradigm. On the other hand, it stands out the role played by Montague in the interest wakened up among the linguists by Categorial Grammars, although it is considered that MG is just a semantic theory.

---

* This paper is appearing in a French version in *Les Grammaires Catégorielles: Histoire et Perspectives contemporaines.* Revue Langages, 146 (2002). Please, refer to that version for citing.

In fact, we can affirm that Richard Montague was the first one to propose a coherent and articulate program applied to a formal conception of syntax and semantics, for what his grammatical project can be considered as the first formal grammar that establishes a relationship between the syntactic and the semantic components of grammar, conceived as the two inseparable faces of a coin.

The semiotic conception of “Universal Grammar” managed by Montague stays in the basis of his grammatical project. For him, universal grammar is not more than a mathematical frame able to describe any definable system of symbols as a language, going from the animal languages to human natural language, passing through formal languages. It is for this he expressly manifests the biggest linking between linguistics and mathematics in front of its relationship with psychology (proposed by Chomsky), because the theoretical difference between natural language and the languages of formal logic is not important:

“There is in my opinion no important theoretical difference between natural languages and the artificial languages of logicians; indeed I consider it possible to comprehend the syntax and semantics of both kinds of languages within a single natural and mathematically precise theory” (Montague 1970, 1974:222).

The publication of the first works by Montague in the fifties coincides in time with Chomsky’s generative theories and the emergence of Transformational Generative Grammar (TGG) (Chomsky 1957). The most relevant fact in connection with this coincidence is that MG supplies a semantic and “pragmatic” analysis of the grammar in front of the purely syntactic analysis of TGG. This probably explains the central role played in the semantic controversies, in general, by logical grammars and, specifically, by MG. This controversy took to the reformulation of the chomskian theory into what has been denominated the “standard theory” of TGG (Chomsky 1965). In the Standard Theory the Katz-Postal thesis is assumed (Katz & Postal 1964) as a formal explanation of the existent relationship between syntax and semantics, quite different, however, of the basic proposals of MG.

The theories of Montague are mainly condensed in three articles published in 1970 -and in that same year’s communication in the Stanford Workshop on Grammar and Semantics about the appropriate treatment of the quantification in English (PTQ)-, just several months before his death (Montague 1970a, 1970b, 1970c, 1973). These theories had an immediate influence in linguistics. In the decade of the seventies, several volumes dedicated to MG appeared. This way, in 1974 the summary of his main articles was published in charge of Richmond H. Thomason (Montague 1974) and two years later Barbara Partee published a volume on the montagovian semiotic program (Partee 1976).

The unconcluded project of Montague was recaptured at once by different logicians and linguists, being carried out some doctoral dissertations as (Bennett 1974) or (Cooper 1975). The comparisons between MG and TGG were also developed in (Partee 1975) and in (Bach 1979). Pragmatic developments of MG were also carried out in this decade, like in (Karttunen & Peters 1978). But without a doubt, what had a more important influence in the knowledge of Montague grammatical program by linguists were the introductions and applications of his theories: (Thomason 1974), (Halvorsen & Ladusaw 1979), (Dowty 1979); and, already in the decade of the 80's, (Dowty, Wall & Peters 1981), where the grammatical theory exposed in PTQ and Intensional Logic -conceived as an intermediate language between formal logic and natural language- are analyzed.

1.2 Some recent developments and applications of Montague Grammar

In the last twenty years, the Montague semiotic program has been developed and applied with different purposes, as much in linguistics as in logic and computational theory. MG has been taken as a reference point in numerous works about categories, semantic theory and discourse interpretation, as well as in some applications of logical grammars in computational linguistics. Let us see briefly some of these works.

In 1986, Johan van Benthem published an interesting book with some essays on logical semantics (Benthem 1986). In this book, logical and grammatical categories are analyzed, as well as determiners, quantifiers, conditionals, tense and modality, using MG as the starting point from which the interpretation of sentence constituents is studied. The concept of *dynamic semantics* (dynamics of interpretation) is introduced in the CGs by means of the application of a Lambek Calculus in MG.

Just a couple of years later, two independent volumes were published with different authors' contributions on Categorial Grammar: (Buszkowski, Marciszewski & Benthem 1988) and (Oehrle, Bach & Wheeler 1988). In both of them, Montague Grammar plays a central role in the application of CGs to the analysis of natural language. Different logical aspects of these grammars are studied, such as the model theoretical component of Categorial Grammars or the implementation of Lambek Calculus applied to categories on the basis of MG, for instance.

In (Benthem 1991) the logical foundations of the CGs are presented in connection with the interface between logic and linguistics (in particular, the Theory of Generalized Quantifiers). For this, van Benthem chooses MG to represent Categorial Grammars, extending it by means of Lambda Calculus and the Theory of Types.

Another interesting point here is the proposed *shift from CG to Dynamic Logic*, motivated by Computer Science and the computational treatment of natural language. The technical setting for this shift lies in Modal Logic and the semantics of possible worlds.

Finally, van Benthem's book deals with the phenomenon of *information processing*, where converge various logical paradigms such as Relevance Logic, Modal Logic or Linear Logic, all of them directed toward the structural and procedural aspects of information.

(Morrill 1994) incides in the semiotic conception of natural language grammar by means of the proposal of a Type Logical Grammar (TLG). The semiotic perspective of MG is opposed to Chomsky's mentalist perspective. In fact, this opposition is in the base of the difference between a real formal logical grammar for natural language and the chomskyan conception of a generative grammar in the present.

The aim of TLG is to generalise CG to a categorial logic. For this, Morrill presents a refinement of Logical Grammar (Logical Syntax and Model-theoretic semantics) "in which 'logical' applies not just to logical semantics, but also to logical types directing derivation". The bases of Type Logical Grammar are:

1. Montague Grammar: a montagovian fragment for English.
2. The Theory of Types
3. A Gentzen Sequent Logic
4. Lambek Calculus (associative and non-associative).
5. Multimodal Systems and Labelled Deduction.

In this sense, TLG offers a wide framework in which some of the applications of natural language logical analysis can be implemented: in the level of semantic interpretation (Carpenter 1997), or in that of discourse interpretation or even in that of the "pragmatic" theories about inferences and semantic enrichment, what gives place to dynamic conceptions of MG.

A frequently visited alternative in the nineties has been the opposition of MG and the dynamic theories of meaning. The montagovian framework has constantly been applied in diverse works related with the theory of meaning and dynamic semantics during the last decade. As an alternative, Discourse Representation Theory (DRT) arose at the beginning of the eighties. It was proposed by Hans Kamp (Kamp 1981, Kamp & Reyle 1993) with the purpose of explaining certain problems of discourse interpretation that escaped to the MG environment.

The necessity to apply dynamic theories of interpretation in the categorial frame was already outlined by Johan van Benthem, as it was seen before. Other authors like (Groenendijk & Stockhof 1987, 1991) have proposed a dynamic treatment of Predicate Logic, enlarged with a typologically structured language, lambda abstraction and Montague Semantics. This is what they call a Dynamic Montague Grammar (DMG).

But maybe the most direct proposal to give a representational treatment of the dynamic discourse processes in MG is (Muskens 1996). The main differences between DRT and MG rest in the fact that DRT is more flexible to represent the discursive relationships; however, the *Compositional Principle*, valid in MG, doesn't hold in DRT. This explains MG biggest elegance when dealing with quantification and coordination phenomena, for example. And the interest to combine the dinamicity of DRT with the compositionality of MG, what constitutes the purpose of Muskens' article.

The underlying logic to DRT is reducible to First Order Predicate Logic. Muskens combines with this a classical Theory of Types Logic, based on Church's Lambda Calculus instead of Montague's Intensional Logic:

"[...] we can have boxes [DRSs] and lambdas in one logic, and the combination of these two [...] will allow us to assign boxes [DRSs] to English discourses in accordance with Frege's Principle: the meaning of a complex expression is a function of the meaning of its parts" (Muskens 1996:144).

At the beginning of the nineties, Dov Gabbay began to develop a general framework for programming and logic: Labeled Deductive Systems (LDS), described in its formal aspects in (Gabbay 1996).

The purpose of LDS is to contribute a unified frame for the main logical theories used in Logic Programming: Relevance Logic, Modal Logic, Linear Logic, Temporal Logic, etc.

In (Gabbay & Kempson 1992, 1996) different aspects of natural language are treated as the interpretation of anaphora, temporal relationships and tense, structural dependences and the logical inferences made in discourse, by means of the implementation in LDS of a Categorial Grammar based on Montague Grammar.

The implementation of MG in LDS offers a powerful tool to analyze what Gabbay and Kempson call "semantic enrichment". Under this denomination we find different linguistic phenomena -syntactic and semantic ones- that intervene in communication, as well as certain pragmatic aspects as the influence of context or of situation in the interpretation of the diverse utterances that conform natural speech. In this sense we can say that LDS becomes a MG-based alternative to the semantics of situations and the semantics of possible worlds as well as to DRT. We will have a look to it.

2. Applying MG to discourse analysis

2.1 The semiotic conception of grammar

As it has been said before, the implementation of MG in LDS permits a categorial dynamic treatment of discourse. This is just one of the different approaches to MG during the last decade of the 20th century; but, in my opinion, the most interesting because of its simplicity and its capacities for modelling linguistic phenomena.

One of the great advantages of CGs -and of MG, in particular- consists in dealing with natural language from a semiotic perspective, thinking on grammar as a device that allows us to distinguish between *grammatical symbol chains* and *non-grammatical symbol chains*. In other words, grammar is only a device to select a subset of symbol chains (those we call *grammatically correct*) from the set of all the possible chains formed by a given alphabet or vocabulary. From a logical point of view, a grammar G of a language L is a set of rules and principles from which it is possible, given a lexicon, to form every correct and meaningful sentence of the language L. So, G can be considered as a deductive system, being L the set of its theorems.

In this definition of grammar, information can be seen as the result of natural language communication processes in which the hearer interprets the speaker's utterances by means of different but related inference tasks: mainly a grammatical and a logical approach to the meaning of utterances. Then, affording the analysis of natural language from a point of view involving grammar as well as logic, makes us to be in disposition of giving a general framework for utterance interpretation.

The grammatical approach involves the recognition of the lexicon and the assignment of syntactic functions to the words. The logical approach allows to establish certain meaning relations among the elements of a speech act such as the words, the context and the logical presuppositions that are assumed by the speaker and the hearer. So we have a complex *continuum* for the interpretation of utterances that leads us from the grammatical relations of the words to the logical deduction of information that is not codified *prima facie* by grammar. We interpret this *continuum* as a system of related databases each of them being a set of lexical entries and its categorial types, and the system itself as a logical device for the interpretation of natural language utterances.

We can consider that lexical entries are the minimal units of information in the linguistic communication process (although this "lexicalist" hypothesis could be discussed). But lexical meaning is not enough to provide an interpretation of sentences. According to the *Principle of Composition*, the meaning of a complex expression is a function of the meanings of its parts and of the syntactic rules by which they are combined. Therefore, the meaning of sentences depends on the meaning of the words that are parts of them plus their combinatorial properties and the contextual system where they are uttered.

The evident cases of "contextual sensitivity" are those sentences where non-referential expressions appear. We call these expressions anaphoric expressions or anaphora. Only by means of logical inferential procedures, the hearer can determine the meaning of sentences of this kind where anaphora plays a role in its interpretation. These inferential procedures depend on certain semantic enrichment processes in which speaker, hearer and context are important elements as well as lexical meaning and system relations. Therefore, explaining the meaning of natural language utterances from a formal perspective requires to develop a logical framework of inference that includes those semantic enrichment processes as a part of the logical inference itself. To put it in a nutshell, we must establish a model of natural language interpretation in terms of natural-deductive reasoning tasks, being necessary to explain certain syntactic and semantic phenomena (v. gr.: the case of anaphora) as logical deductive processes.

2.2 A logical device for utterance interpretation.

Labelled Deductive Systems (LDS) are a general computational framework for inferential processes (Gabbay 1996). Its application to natural language analysis as in (Gabbay & Kempson 1992) provides a set of databases whose data are lexical entries (labels), everyone of them being followed by an assigned categorial functional type that can be interpreted in a logical way since functional types behave very much like logical implications (Benthem 1991:35). Each database is labelled itself, so it is possible to construct a map of related databases through some labelling functions.

In more strict terms, an LDS is a pair $<L,\Gamma>$, where L is a logic and Γ is an algebra with some operations on labels. The chosen logic L for the analysis of natural language is the implicational fragment of a relevance linear propositional logic such that a *well formed formula* (wff.) of logic L is a formula that fulfills the following conditions:

1. *e* and *t* are wffs.
2. If A and B are wffs. then $A \rightarrow B$ is a wff.
3. Nothing more is a wff.

A *correct expression* (or simply an *expression*) is a pair α:A, where $\alpha \in \Gamma$ and A is a wff. The labels of the set Γ can be lexical entries of a certain natural language (v.gr.: English or Spanish) and the wff. A, the functional type corresponding to that lexical entry. A set of expressions forms a *database*.

A *deduction* Δ in LDS consists in getting from some assumptions of the form α_1:A_1,...α_n:A_n, where α_1,...α_n are labels and A_1,...A_n are formulae, an expression of the form $\phi(\alpha_1,...\alpha_n)$:t, where $\phi(\alpha_1,...\alpha_n)$ is a label obtained by combinatorial processes of α_1,...α_n and t is the formula representing the categorial type *t* obtained from A_1,...A_n by means of some rules defined in a natural deduction way.

We say a sentence of a natural language *S* is *derived* in LDS when we show a deduction Δ in LDS such that its last expression is of the form $\phi(\alpha_1,...\alpha_n)$:t, where $\phi(\alpha_1,...\alpha_n)$ is the formal functional counterpart of *S*.

Every assumption must be used in a deduction in LDS (relevance requisite) and they must be used only once (linearity requisite). When we derive by deduction in a database δ_k an expression of the form α:t and the relevance and linearity requisites are fulfilled, we say that δ_k is closed. It is also possible to open a new database in any moment of the deduction just making an assumption. This new database will be nested in the previous open database.

Several databases can be related in a deduction. The relation is similar to the accessibility relation among model sets in modal logic (Salguero 1991:57-60). Let $\Re$ be such a relation. Let $D_e(\delta_k)$ be the domain of the database δ_k, that is to say, the set of referents of the expressions of type *e* in δ_k. We say that for every two databases related by $\Re$, the population of the first database is inherited by the second one. In symbols:

$$\forall \delta_i \delta_k [\delta_i \Re \delta_k \Rightarrow D_e(\delta_i) \subseteq D_e(\delta_k)]$$

This *requisite of nested domains* applied to LDS databases will be very useful for certain cases of anaphora, as we will see in the next section.

The basic rules of LDS are the following:

R1. Application: For every two expressions α:A$\rightarrow$B$\in \delta_k$ and β:A$\in \delta_n$, we can add to the actual database δ_i an expression of the form $\alpha(\beta)$:B iff either $\delta_k \Re \delta_n \Re \delta_i$ or $\delta_n \Re \delta_k \Re \delta_i$.

R2. λ-Abstraction: For every two expressions $x{:}A \in \delta_n$ and $\alpha(x){:}B \in \delta_n$, where x:A is the only assumption of δ_n and α(x):B has been derived in δ_n, if $\delta_k \Re \delta_n$ then $\lambda x \alpha(x){:}A \rightarrow B \in \delta_k$ and δ_n is closed.

R3. λ-Conversion: If $\lambda x[\alpha(x)](\beta){:}A \in \delta_k$ then $\alpha(\beta){:}A \in \delta_k$.

R4. Reutilization: For every expression $\alpha{:}A \in D_e(\delta_k)$ we can add this expression to another database δ_n iff $\delta_k \Re \delta_n$.

The rules of Application and λ-Abstraction are the labelled forms of classical propositional logic rules of elimination (Modus Ponens) and introduction of the implicational connective. Nevertheless, the rule of λ-Conversion is a rule operating only in the label, leaving the formula in the expression untouched.

Databases so conceived are in many aspects like possible worlds. For example, we can see them as moments, interpreting databases in a temporal manner. Or, much better, as states of knowledge of the speaker/hearer of a natural language sentence, almost like an *information state* in the kripkean sense of the term. This makes possible to interpret the expressions in a database like the theorems that hold in a model set (Hintikka set). Therefore, for every expression $\alpha_i{:}A_i$ appearing in a database δ_k, we will say that $\delta_k \vdash \alpha_i{:}A_i$.

It is important the characterization of the relation ℜ for making the system more or less powerful. The best characterization of ℜ for our proposals is as a partial order, viz.: ℜ is a reflexive, antisymmetric, transitive and connected relation. This makes the concept of LDS databases closer to the kripkean concept of *information state*. So, a database contains not just the information of the expressions belonging to it, but also the information derived from the expressions belonging to "previous" information states and the information to be derived in the database itself.

So, what we have is that LDS allows to construct a database where all the lexical information of a natural language available in a certain moment for a speaker/hearer is put. This information is mainly a label and a functional type (and maybe an ordering) for each lexical entry. So, given such a database, it is possible to manipulate the information into it with rules defined in function of the needed logic in order to increase the database. As every functional type is interpreted as a categorial logical type, LDS provides a parser that assigns a structured database to every grammatically correct sequence of words as its interpretation.

2.3 The treatment of anaphoric expressions: context and presuppositions

We consider an anaphora is any expression in the discourse whose meaning does not depend on the expression itself, but on another expression in the discourse to which the anaphora is (grammatically) related. The scope where an anaphora finds its reference does not necessarily correspond to the sentence in which it appears. Moreover, anaphoric expressions usually find their reference in other sentences or even in extralinguistic context presuppositions.

There are different types of anaphora. When we talk about anaphora, we can be talking about pronominal anaphora, relative clauses, definite and indefinite noun phrases, verb phrase anaphora, tense (and aspect) anaphora or, even, ellipsis. However, in this paper we will only deal with some types of anaphora. On the one hand we have those anaphoric expressions like pronouns, relative clauses and noun phrases, expressions that look for their reference in other expressions in the discourse provided a certain kind of quantification is involved. On the other hand we have verb phrase anaphora, tense, aspect and ellipsis, where the problem is not variable instantiation, at least not in the same sense. Let us analyze the former ones.

We can distinguish several kinds of pronominal anaphora. For example, we have correferential pronouns as in the sentence:

(1) John loves Mary. *She* hates *him*.

The simple logical form of (1) is:

[1] $love(m,j) \wedge \exists xy hate(y,x)$

Applying an LDS analysis to (1) we have two related databases δ_1 and δ_2 such that δ_1=<john':e, love':e→(e→t), mary':e> and δ_2=<x_{she}':e hate':e→(e→t), y_{him}':e>. If we apply the rules of our calculus to δ_1 we get:

$$\delta_1 = \langle john':e,\ love':e \to (e \to t),\ mary':e,\ love(mary)':e \to t,\ love(mary)(john)':t \rangle$$

But, what about applying the rules of the calculus to δ_2? Remember we have got a quantified formula as its logical form. This means we must care about the existential presuppositions that lie under the sentence "She hates him" to get its interpretation.

As we have seen, LDS databases are like possible worlds in several aspects. So we can treat them as though they were model sets. To these model sets we can apply certain *individuating functions* that preserve the reference of variables in intensional contexts (Salguero 1991:130-139).

Moreover, we can add certain marks to the labels of our assumptions to preserve certain referential aspects of lexicon as gender, for instance. These marks on the labels in a deduction can be treated as individuating functions applied to the lexicon. For example, let D_e be the set of denotations of all the expressions of type *e* in English (the domain of discourse), being $D_e(\delta_k)$ a proper subset of D_e whose members are the expressions of type *e* that appear in the database δ_k (the population of δ_k) such that for every database δ_n, if $\delta_k\Re\delta_n$ then $D_e(\delta_k)\subseteq D_e(\delta_n)$ by the *requisite of nested domains* and let *f* be the individuating function for feminine words in English. This function assigns to every expression $\alpha\in D_e$ the value 1 if and only if α is a lexical feminine entry in an English lexicon. Otherwise its value is 0. Then we can add a restriction to the reutilization rule imposing the condition for the instantiation of a λ-bound variable x that $f(x\in D_e(\delta_n))=f(\alpha\in D_e(\delta_k))$ and $\delta_k\Re\delta_n$. In our example, $f(x_{she}\text{'}:e\in\delta_2)=mary\text{'}:e\in\delta_1$, $f(y_{him}\text{'}:e\in\delta_2)=john\text{'}:e\in\delta_1$ and $\delta_1\Re\delta_2$.

By applying this restriction and the rules of the calculus to δ_2 we obtain:

$$\delta_2=<x_{she}\text{'}:e\ hate\text{'}:e\rightarrow(e\rightarrow t),\ y_{him}\text{'}:e,\ f(x_{she})=mary\text{'}:e,\ f(y_{him})=john\text{'}:e,\\ hate(john)\text{'}:e\rightarrow t,\ hate(john)(mary)\text{'}:t>$$

This database is an abbreviated way to write the whole database δ_2, where the individuating function *f* introduces a λ-abstraction process in the derivation as follows:

$$\delta_2=<x_{she}\text{'}:e\ hate\text{'}:e\rightarrow(e\rightarrow t),\ y_{him}\text{'}:e,\ \delta_3,\\ \lambda x_{she}[\lambda y_{him}[hate(y_{him})](x_{she})]\text{'}:e\rightarrow(e\rightarrow t),\ mary\text{'}:e,\\ \lambda x_{she}[\lambda y_{him}[hate(y_{him})](x_{she})](mary)\text{'}:e\rightarrow t,\\ \lambda y_{him}[hate(y_{him})(mary)]\text{'}:e\rightarrow t,\ john\text{'}:e,\\ \lambda y_{him}[hate(y_{him})(mary)](john)\text{'}:t,\ hate(john)(mary)\text{'}:t>$$

where δ_3 is:

$$\delta_3=<x_{she}\text{'}:e,\ \delta_4,\ \lambda y_{him}[hate(y_{him})]\text{'}:e\rightarrow(e\rightarrow t),\\ \lambda y_{him}[hate(y_{him})](x_{she})\text{'}:e\rightarrow t>$$

where δ_4 is:

$$\delta_4=<y_{him}\text{'}:e,\ hate(y_{him})\text{'}:e\rightarrow t>$$

and $\delta_2\Re\delta_3\Re\delta_4$. For a more detailed discussion see (Salguero 1994).

A second type of pronominal anaphora are pronominal variables bound by a quantifier. The following sentence is an example:

(2) Every student is proud of *his* work.

We can get two different logical forms of (2) related to its two different interpretations:

[2] $\forall x \exists y(\text{student}(x) \wedge \text{work}(y) \wedge \text{belong}(x,y) \rightarrow \text{be\_proud}(y,x))$

[2’] $\forall x \exists yz(\text{student}(x) \wedge \text{work}(y) \wedge \text{belong}(z,y) \rightarrow \text{be\_proud}(y,x))$

In both cases, the reference of the anaphoric expression “his” depends on a quantifier, either a universal or an existential one. The reference of the anaphora is a function again, but this time the function takes its value from the whole domain the quantifiers are operating over:

$$f(x_{his} \in D_e(\delta_1) = g(\{w \mid w \in D_e\})$$

The difference between the interpretations [2] and [2’] is a certain restriction on the function *g*. While interpretation [2] requires that $\|\lambda x[\text{student}(x)](a)\|=1$ for every $a \in D_e$ such that $g(w)=a$, interpretation [2’] only requires that $D_e \neq \emptyset$. That is to say, [2] requires a real individuating function but [2’] only an existential presupposition.

The treatment of relative clauses is similar to the treatment of pronominal anaphora. The sentence

(3) John loves Mary who hates him.

is analyzed in LDS in the same way we analyzed the sentence (1). The only difference is that in the analysis of (1) we have got two main related databases δ_1 and δ_2 and a number of nested databases in δ_2 obtained from several processes of λ-abstraction, while in the analysis of (2) we have a single set of nested databases: its derivation is very similar to a natural calculus derivation with several auxiliary hypothesis. The restrictions applied in the analysis of (1) are applied in the analysis of (3) to obtain the reference of the anaphoric expressions.

An important type of pronominal anaphora are the well known indirectly bound pronouns whose best examples are the “donkey sentences”:

(4) Everyone *who* owns *a donkey* beats *it*.

Its logical form is:

[4] $\forall xy(\text{donkey}(y) \wedge \text{own}(y,x) \rightarrow \text{beat}(y,x))$

In (4) we have a similar case to the previous ones. Its peculiarity consists in that the reference of the anaphoric expression “it” depends on the reference of the indefinite noun phrase “a donkey” in the same database, whose reference depends on the relative “who”, whose reference depends on the quantifier “everyone”. So, we have a very good example of a complex process of inference from logical instantiation of non-referential variables.

The behavior of an indefinite noun phrase as “a donkey” is somehow identical to the behavior of a quantified predicative sentence. The same is true for definite noun phrases as “the donkey”. In [4] we have a universal quantification of the anaphoric variable induced by the universal quantification under whose scope the noun phrase is. It would have been possible the alternative existential logical analysis:

[4’] $\forall x \exists y(donkey(y) \wedge own(y,x) \rightarrow beat(y,x))$

In any case, we have got a problem of existential presupposition as we had in the analysis of (2) above. Therefore, the peculiar problems that arise from the analysis of indirectly bound pronouns are treated in LDS as a set of instantiation tasks of anaphoric expression as we did in (1) and (2), and the whole problem is reduced to the definition of the corresponding individuating functions.

2.4 The treatment of anaphoric expressions: pronominal relations, relative clauses and dependencies

LDS explains in a very natural way certain phenomena related to the use of pronouns. This is the case of object reduplication in some romance languages like Italian or Spanish. Consider, for example, the following Spanish sentence

(5) Juan le dio una rosa a María

John her-Dat. gave a rose-Ac. Mary-Dat.

“John gave a rose to Mary”

The personal pronoun “le” is related to “María”. Both words are in dative case while “una rosa” is accusative, playing the role of the direct object of the transitive verb “dio”. The question is: have we got in the sentence (5) two indirect objects or just one splitted indirect object? It is hard to see the phenomenon as a splitted object since the following two sentences are both grammatically correct in Spanish:

(6) Juan le dio una rosa

“John gave her a rose”

(7) Juan dio una rosa a María

“John gave a rose to Mary”

So it is a case of double indirect object, only possible because the reference of the anaphora “le” is the same individual than the reference of the name “María”: the person we call Mary. The derivation of (5) in LDS is not a problem:

$$\delta_1 = <\text{Juan}':e,\ x_{le}':e,\ \text{dar}':e\rightarrow(e\rightarrow(e\rightarrow t)),\ \text{una\_rosa}':e,\ \text{a\_María}':e,\ \delta_2,$$
$$\lambda x_{le}[\text{dar}(\text{una\_rosa})(x_{le})]':e\rightarrow(e\rightarrow t),$$
$$\lambda x_{le}[\text{dar}(\text{una\_rosa})(x_{le})](\text{a\_María})':e\rightarrow t,\ \text{dar}(\text{una\_rosa})(\text{a\_María})':e\rightarrow t,$$
$$\text{dar}(\text{una\_rosa})(\text{a\_María})(\text{Juan})':t>$$

where δ_2 is

$$\delta_2 = <x_{le}':e,\ \text{dar}(\text{una\_rosa})':e\rightarrow(e\rightarrow t),\ \text{dar}(\text{una\_rosa})(x_{le})':e\rightarrow t>$$

In this derivation we do not need use the rule of reutilization because its role is played in the sentence itself by the duplicated object. Then, the name “María” is directly instantiating the anaphora “le” in the same database without any outer reference being needed.

Relatives are well known anaphoric expressions too. In many aspects, they are like pronominal anaphora, but there is an important characteristic that makes them to be different: relative clauses introduce a subordinate sentence, so they are a part of a different database than the rest of the sentence where its antecedent is. Therefore, relative anaphora is a control label that ever imposes the opening of a new nested database in a LDS deduction. This database is closed when we reach an expression of type *t* as usual. Let the following sentence be an example:

(8) Mary loves John who hates her

Its deduction in LDS is very similar to the previous ones except for the fact that the relative “who” works as a new database control:

$$\delta_1 = <\text{Mary}':e,\ \text{love}':e\rightarrow(e\rightarrow t),\ \text{John}':e,\ \delta_2,\ \text{love}(\text{John})':e\rightarrow t,$$
$$\text{love}(\text{John})(\text{Mary})':t>$$

where δ_2 is

$$\delta_2 = <x_{who}':e,\ \text{hate}':e\rightarrow(e\rightarrow t),\ x_{her}':e,\ \delta_3,$$
$$\lambda x_{who}[\text{hate}(\text{Mary})(x_{who})]':e\rightarrow t,\ \text{John}':e,$$
$$\lambda x_{who}[\text{hate}(\text{Mary})(x_{who})](\text{John})':t,\ \text{hate}(\text{Mary})(\text{John})':t>$$

and δ_3 is

$$\delta_3 = <x_{who}':e,\ \delta_4,\ \lambda x_{her}[\text{hate}(x_{her})]':e\rightarrow(e\rightarrow t),\ \text{Mary}':e,$$
$$\lambda x_{her}[\text{hate}(x_{her})](\text{Mary})':e\rightarrow t,\ \text{hate}(\text{Mary})':e\rightarrow t,\ \text{hate}(\text{Mary})(x_{who})':t>$$

and δ_4 is

$$\delta_4 = <x_{her}':e,\ \text{hate}(x_{her})':e\rightarrow t>$$

This treatment of the relative clauses differs from the treatment in (Gabbay & Kempson, 1992). Like them, we consider the anaphora “x_{who}” is a control label that tell us it is necessary to open a new database. But this database is nested in the original one, not linked to it by a function on the antecedent of the anaphora. Otherwise, we could not save the order of words in the deduction of the following sentence:

(9) John, who hates Mary, loves Susan

The derivation of (9) in LDS is easily comparable to the derivation of (8):

$$\delta_1 = \langle \text{John'}:e,\ \delta_2,\ \text{love'}:e\rightarrow(e\rightarrow t),\ \text{Susan'}:e,\ \text{love(Susan)'}:e\rightarrow t,\ \text{love(Susan)(John)'}:t \rangle$$

where δ_2 is

$$\delta_2 = \langle x_{who}\text{'}:e,\ \text{hate'}:e\rightarrow(e\rightarrow t),\ \text{Mary'}:e,\ \delta_3,\ \lambda x_{who}[\text{hate(Mary)}(x_{who})]\text{'}:e\rightarrow t,\ \text{John'}:e,\ \lambda x_{who}[\text{hate(Mary)}(x_{who})]\text{(John)'}:t,\ \text{hate(Mary)(John)'}:t \rangle$$

and δ_3 is

$$\delta_3 = \langle x_{who}\text{'}:e,\ \text{hate(Mary)'}:e\rightarrow t,\ \text{hate(Mary)}(x_{who})\text{'}:t \rangle$$

This treatment allows us to maintain the order of words in the deduction, deriving in the first place the relative subordinate sentence and then the main one, what is closer to the process of information decodification by the hearer than deriving the main sentence first and then the subordinate one as a linked database.

References

Bach, E. (1979). “Montague Grammar and classical Transformational Grammar”, in Davies & Mithun (eds.) (1979).

Bennett, M. (1974). *Some Extensions of a Montague Fragment*. PhD dissertation. UCLA.

Benthem, J. van (1986). *Essays in logical semantics*. Dordrecht: Reidel.

Benthem, J. van (1991). *Language in action. Categories, lambdas and dynamic logic*, Amsterdam: North-Holland.

Buszkowski, W., W. Marciszewski & J. Van Benthem (eds.) (1988). *Categorial grammar*. Amsterdam: John Benjamins.

Carpenter, R. (1997). *Type-Logical Semantics*. Cambridge (Mass.): MIT Press.

Chomsky, N. (1957). *Syntactic structures*. La Haya: Mouton.

Chomsky, N. (1965). *Aspects of the theory of syntax*. Cambridge (Mass.): MIT Press.

Cooper, R. (1975). *Montague's Semantic Theory and Transformational Syntax*. PhD. dissertation. University of Massachusetts at Amherst.

Davies, S. & Mithun, M. (eds.) (1979). *Linguistics, Philosophy and Montague Grammar*. Austin: University of Texas Press.

Dowty, D. (1979). *Word meaning and Montague Grammar. The semantics of verbs and times in generative semantics and in Montague's PTQ*. Reidel, Dordrecht.

Dowty, D., R. E. Wall, & S. Peters (1981). *Introduction to Montague Semantics*. Dordrecht: Reidel.

Gabbay, D. (1996). *Labelled Deductive Systems*. Oxford: The Clarendon Press.

Gabbay, D. & R. Kempson (1992). "Natural-Language content: a truth theoretic perspective". *Proceedings of the 8th Amsterdam Formal Semantics Colloquium*, University of Amsterdam.

Gabbay, D. & R. Kempson (1996). "Language and proof theory". *Journal of Logic*, *Language and Information*, 5/3-4: 247-251.

Gazdar, G. & G. Pullum, (1985). "Computationally relevant properties of natural languages and their grammars". In W. J. Savitch & al. (eds.): *The formal complexity of natural language*. Dordrecht: Reidel, 1987:387-437.

Groenendijk, J. & M. Stockhof (1987). "Dynamic Montague Grammar". In L. Kálman & L. Pólos (eds.), *Papers from the Second Symposium on Logic and Language*, Budapest: Akadémiai Kiadó, 1990, páginas 3-48.

Groenendijk, J. & M. Stockhof (1991). "Dynamic predicate logic". *Linguistics and Philosophy*, 14/1: 39-100.

Halvorsen, P. K. & W. Ladusaw (1979). "Montague's 'Universal Grammar':An introduction for the linguist". *Linguistics and Philosophy*, 3:185-223.

Kamp, H. (1981). "A theory of truth and discourse representation", en Groenendijk, J., T. Jansen & M. Stockhof (eds.), *Formal methods in the study of language*. Amsterdam: Mathematical Centre tracts 135, 1981.

Kamp, H. & U. Reyle (1993). *From Discourse to Logic*, Dordrecht: Kluwer.

Karttunen, L. & S. Peters (1978). “Conventional implicature in Montague Grammar”. *Proceedings of the First Annual Meeting of the Berkeley Linguistic Society*, pp. 266-278.

Katz, J. J. & P. Postal (1964). *An integrated theory of linguistic descriptions.* Cambridge (Mass.): MIT Press.

Montague, R. (1970a) “Pragmatics and Intensional Logic”. *Synthèse*, 22:68-94.

Montague, R. (1970b) “English as a Formal Language”. In B. Visentini & al. (eds.), *Linguaggi nella Società e nella Tecnica*, Milán: Edizioni di Comunità, pp. 189-224.

Montague, R. (1970c) “Universal Grammar”. *Theoria*, 36:373-398.

Montague, R. (1973) “The Proper Treatment of Quantification in Ordinary English”. In J. Hintikka, J. Moravcsik & P. Suppes (eds.), *Approaches to Natural Language: Proceedings of the 1970 Stanford Workshop on Grammar and Semantics,* Dordrecht: Reidel, 1973, pp. 221-242.

Montague, R. (1974). *Formal philosophy*. *Selected papers of Richard Montague.* New Haven/London: Yale U. P.

Morrill, G. V. (1994). *Type logical grammar*. *Categorial logic of signs.* Dordrecht: Kluwer.

Muskens, R. (1996). “Combining Montague semantics and discourse representation”. *Linguistics and Philosophy*, 19/2: 143-186.

Oehrle, R. T., E. Bach & D. Wheeler (eds.) (1988). *Categorial grammars and natural language structures.* Reidel, Dordrecht.

Partee, B. (1975). “Montague Grammar and Transformational Grammar”. Linguistic Inquiry, 6:203-300.

Partee, B. (ed.) (1976). *Montague Grammar*. Nueva York: Academic Press.

Salguero, F. J. (1991): *Árboles semánticos para lógica modal con algunos resultados sobre sistemas normales.* PhD dissertation. University of Seville.

Salguero, F. J. (1994): “Anaphoric instantiation problems in an inference model of utterance representation”. In Carlos Martín-Vide (ed.): *Current issues in Mathematical Linguistics*. Amsterdam: North Holland, 1994:39-48

Thomason, R. (1974): “Introduction”, en R. Montague (1974).