Francisco J. Salguero Lamillar
Área de Lingüística General
Universidad de Sevilla
E-mail: salguero@cica.es

REFERENCE WITHIN DEDUCTIVE PROCESSES IN COMMUNICATION: THE MEANING OF NON-REFERENTIAL EXPRESSIONS

The recent development of the categorial framework as a tool for the analysis of natural language implies the acceptance of the application in linguistics of formal procedures coming from logic and computation theory. The classical problems related to the syntax-semantics interface, for instance, can be treated in categorial grammar better than in other grammatical theories. In this sense, one of the most relevant tasks for the grammarian is conjugating the categorial format of montagovian intensional languages and Lambek calculi to obtain "interpreted" categorial calculi as powerful as to establish at the same time the syntactic and semantic correctness of the sentences of a natural language. This means that such calculi must assign to every correct syntactic structure a semantic interpretation and *vice versa*, becoming syntax and semantics the two indivisible sides of a same coin.

1. Grammar and the meaning of non-referential expressions

We can define 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 (Partee & al. 1990:433-439).

From 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, as we do, makes us to be in disposition of giving a general framework for utterance interpretation, which is the aim of this paper.

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 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.

For the purposes of this paper, we will consider that lexical entries are the minimal units of information in the linguistic communication process. But the 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 (Partee & al. 1990:318). 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 case are those sentences where non-referential expressions appear. We call these expressions anaphoric expressions or anaphora. Then, in this paper we will 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.

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 from a formal perspective the meaning of natural language utterances 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. A logical device for utterance interpretation

Labelled Deductive Systems (LDS) are a general computational framework for inferential processes (Gabbay 1991). 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 (van 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 $\alpha_1:A_1,...\alpha_n:A_n$, where $\alpha_1,...\alpha_n$ 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 $\alpha_1,...\alpha_n$ 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 $D_e(\delta_k)$ 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→B∈δ_k and β:A∈δ_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∈δ_n and α(x):B∈δ_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→B∈δ_k and δ_n is closed.

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

R4. Reutilization: For every expression α:A∈$D_e(\delta_k)$ 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 α_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 $\Re$ for making the system more or less powerful. The best characterization of $\Re$ for our proposals is as a partial order, viz.: $\Re$ 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.

3. The meaning of non-referential expressions: pronouns, relatives and noun phrases

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 one of 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:

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

The logical form of (1) is:

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

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:

δ_1=<john':e, love':e→(e→t), mary':e, love(mary)':e→t, love(mary)(john)':t>

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. (We do not care about the distinction between masculine and neuter genders by now). 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, $f(x_{she}':e\in\delta_2)=mary':e\in\delta_1$, $f(y_{him}':e\in\delta_2)=john':e\in\delta_1$ and $\delta_1\Re\delta_2$.

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

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

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:

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

where δ_3 is:

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

where δ_4 is:

$$\delta_4 = \langle y_{him}' : e,\ hate(y_{him})' : e \rightarrow t \rangle$$

and $\delta_2 \Re \delta_3 \Re \delta_4$. For a more detailed discussion see (Salguero 1993).

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

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

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

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

$$\forall x \exists yz(student(x) \wedge work(y) \wedge belong(z,y) \rightarrow be\_proud(y,x)) \quad [2']$$

In both cases, the reference of the anaphoric expression "his" depends on a quantifier, either an 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 $||\lambda x[student(x)](a)||=1$ 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

John loves Mary who hates him. (3)

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":

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

Its logical form is:

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

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:

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

In any case, we have got a problem of existential presupposition as we had in the analysis of (2) before. 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.

REFERENCES

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

Gabbay, D. 1991. *Labelled Deductive Systems*. London: Imperial College, ms. (To be published in Oxford: Oxford University Press).

Gabbay, D. & Kempson, R. 1992. "Natural-Language content and information flow: a proof-theoretic perspective". London: Imperial College and School of Oriental & African Studies, ms.

Partee, B., Meulen, A. ter & Wall, R. 1990. *Mathematical methods in linguistics*. Dordrecht: Kluwer.

Salguero, F. 1991. *Árboles semánticos para lógica modal con algunos resultados sobre sistemas normales.* Sevilla: Universidad de Sevilla, Ph. D. dissertation.

Salguero, F. 1993. "Anaphoric instantiation problems in an inferential model of utterance representation". To be published in C. Martín-Vide, ed., *Current issues in mathematical linguistics*. Amsterdam: North Holland.