STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **XLV**, Number 1, 2001

INTENSIONAL LOGIC TRANSLATION FOR QUANTITATIVE NATURAL LANGUAGE SENTENCES

ADRIAN ONEŢ, DOINA TĂTAR

Abstract. The performance of some natural language processing tasks improves if semantic processing is involved. Moreover, some tasks (database query) cannot be carried out at all without semantic processing. The first semantic description was developed by Montague and all later approach to semantic in the frame of discourse representation theory follow Montague in using more powerful logic language. The present paper is a contribution in treatment of quantitative natural sentences.

1. Introduction

At present, there doesn't exists a general theory of the semantics of natural language. A rigorous analysis of natural language can't be realized without the intensional logic introduced in computational linguistic by Montague. The intensional logic is a further development from the model provided by first order logic. In his semantic analysis of a sentence, Montague distinguishes two elements: intension (or sense) and extension (or reference). The intension of a sentence is even the proposition it expresses, and the extension is its truth value. An extensional logic can only assign truth value to sentences while an intensional logic can, in addition, assign a meaning to these sentences. Moreover, Montague saw in type theory a powerful system which could correspond to the system of syntactic categories of a natural language.

The following sections aim at acquainting the reader with the fundamentals of intensional logic. The last section introduces some proposals in treatment of quantitative natural sentences.

2. Intensional logic

The intensional logic [7] contains, besides new ones, all the concepts in a predicate logic of first order, L_1.

2000 *Mathematics Subject Classification.* 68U35.
1998 *CR Categories and Descriptors.* I.2.1 [**Computing Methodologies**]: Artificial Intelligence – *Appplications and Expert Systems.*

2.1. **A type-theoretic version of the language L_1.** Let t and e be two basic symbols that respectively represent 'truth' and 'entity'. The set of types is recursively defined as follows:

Definition:

- t and e are types (base);
- If a and b are types, then $\langle a, b\rangle$ is a type;
- All types are obtained by applying the base and induction rule a finite number of times.

Example:

$\langle e, \langle t, e\rangle\rangle$ is a type.

In the following the symbol D_a will represent the set of all denotations for the expressions of type a, with respect to a given interpretation domain A. The sets D_a are recursively defined as:

- $D_e = A$;
- $D_t = 0, 1\, or\, T, F$;
- $D_{\langle a,b\rangle} = the\, set\, of\, functions\, from\, D_a\, to\, D_b$.

If the type is $\langle a_1, \langle a_2, \langle \cdots \langle a_n, b\rangle \cdots ..\rangle$ then $D_{\langle a_1,\rangle a_2, \langle \cdots \langle a_n, b\rangle \cdots ..\rangle}$ is the set of functions from $D_{a_1} \times D_{a_2} \cdots \times D_{a_n}$ to D_b.

The syntactic categories are set of expressions in this logic language. We shall assign to each of these categories a syntactic label (a type). The rule of this labeling is denoted *cancellation rule* and can be stated formally as follows [7]:

If α is an expression of type $\langle a, b\rangle$ and β is an expression of type a then the juxtaposition $\alpha(\beta)$ is of the type b.

Let us remark that the rule is conform with the denotation of types: *If α is a function from D_a to D_b and β is an element from D_a then juxtaposition $\alpha(\beta)$ is an element from D_b.*

We can now define the language L_1 *with types*, denoted L_t, as follows:

Definition:

- The type of the constants c_i, and variables x_j is e;
- The type for one-place predicate constant P_i is $\langle e, t\rangle$ (a function from D_e to $D_t = \{T, F\}$);
- The type for two-place predicate constant P_i is $\langle e, \langle e, t\rangle\rangle$ (a function from $D_e \times D_e$ to $D_t = \{T, F\}$);
- The type for n-place predicate constant P_i is $\langle e, \langle e, \cdots \langle e, t\rangle \cdots\rangle\rangle$, with n occurrences of e; (a function from $D_e \cdots \times D_e$ to $D_t = \{T, F\}$);
- If α is an expression of type $\langle a, b\rangle$ and β is an expression of type a then the juxtaposition $\alpha(\beta)$ is of the type b.
- The type of the formulas is t;
- The type of the connective $\neg$ is $\langle t, t\rangle$;
- The type of the connectives $\wedge, \vee, \rightarrow$ is $\langle t, \langle t, t\rangle\rangle$;

- If A is a formula (of type t) and x is a variable (of type e), then $[\forall x A]$, $[\exists x A]$ are of type t.

2.2. **Lambda calculus.** The language L_t will be expanded by adding the lambda operator (λ-operator). The lambda calculus was introduced in the linguistic's community by Montague, and in the logic by Alonzo Church (1941). Replacing the well known notation for the sets as: $\{x|x \textit{ has a certain propriety}\}$ the following notation is used: $\lambda x[formula\ containing\ x]$. The expression

$$\lambda x[formula\ containing\, x]$$

is called λ-expression or λx-abstraction.

The type of a λ-expression as above is $\langle e, t\rangle$. If we shall combine this expression with a constant (of type e), from the cancellation rule results a formula (of type t. This process is named λ-conversion and may be written as follows:

$$\lambda x[\cdots x \cdots](\alpha) = [\cdots \alpha \cdots].$$

In the formula $[\cdots \alpha \cdots]$ each free occurrence of x is replaced with α, the result is $[\cdots \alpha \cdots]$.

Most of the time, in the language L_t, x and A can be of types more general than e and t. The rule is:

If α is an expression of the type a and x is a variable of the type b, then $\lambda x[\alpha]$ is an expression of type $\langle b, a\rangle$.

For example the type of x can be $\langle e, t\rangle$, and in this case λ- operator make a λ abstraction after a predicate (not a variable).

Montague gave some examples of λ-operator in natural language sentences in English [7]. Let us consider two sentences: *Every student walks* and *Every students reads*. Their usual translation in predicate logic is:

$\forall x(S(x) \to W(x))$ and $\forall x(S(x) \to R(x))$.

These sentences are instances of a more general sentence whose translation is a second-order logic formula, i.e. they are λ-conversions of the λ-expression:

$$\lambda Y[\forall x(S(x) \to Y(x))].$$

The first conversion is

$$\lambda Y[\forall x(S(x) \to Y(x))](W)$$

and the second one is

$$\lambda Y[\forall x(S(x) \to Y(x))](R).$$

Therefore, the λ -expression $\lambda Y[\forall x(S(x) \to Y(x))]$ will have the type $\langle\langle e, t\rangle, t\rangle$ and it is equivalent to the English sentence *every student*.

Analogously, we can obtain a λ expression for *every*: if *every student* is $\lambda Y[\forall x\ (S(x) \to Y(x))]$, then this expression may be seen as conversion of the more general expression: $\lambda Z[\lambda Y[\forall x(Z(x) \to Y(x))]]$ for S, that means $\lambda Z[\lambda Y[\forall x(Z(x) \to$

$Y(x))]](S)$. So *every* can be translated in $\lambda Z[\lambda Y[\forall x(Z(x) \to Y(x))]]$ with the type $\langle\langle e,t\rangle, \langle\langle e,t\rangle, t\rangle\rangle$.

Analogously, *some student* or *a student* is translated in expression $\lambda Y[\exists x(S(x) \wedge Y(x))]$ with the type $\langle\langle e,t\rangle, t\rangle$, and *some*, *a*, *an*, in expression $\lambda Z[\lambda Y[\exists x(S(x) \wedge Y(x))]]$ with the type $\langle\langle e,t\rangle, \langle\langle e,t\rangle, t\rangle\rangle$.

2.3. **Intensionality.** In his theory Montague make a distinction between the *sense* (intention) of an expression and $reference$ (extension): the $reference$ of an expression corresponds to semantic (truth) value of this expression, the *sense* corresponds to the meaning of the expression. The distinction between *sense* and $reference$ is important when the operators $\Box$ (necessarily) and $\Diamond$ (possibly) are used. For example, $\Box A$ cannot be described as function of the references of its parts ($\Box$) and A) but can be described as a function of the senses of these parts. The intensionality in natural language is induced by propositional attitude verbs as: *think, believe, regret*, etc.

In the following we shall denote by α^i and α^e, *intension* and *extension* of an expression α, respectively. There is a *cancellation rule* ^{e,i} very important in simplification of the expressions produced by the translation of sentences in natural language: $\alpha^{i,e} = \alpha^{e,i} = \alpha$.

The expressions for determinants *every* and a, *some* will be, considering the intensionality:

$$\lambda Z[\lambda Y[\forall x(Z^e(x) \to Y^e(x))]]$$

respectively

$$\lambda Z[\lambda Y[\exists x(Z^e(x) \wedge Y^e(x))]].$$

As in the case of type-theoretic version of the language L_1, L_t, the syntax contains recursive definitions of the types. The basic types are, as above t, e . Additionally, is used the symbol s for *sense*, which will allow to associate with every type a a new type $\langle s, a\rangle$.

Observation:

The expression of type $\langle s, a\rangle$ have as extension, intension of expression of type a.

The formation rules for types are as follows:

Definition:

- t and e are types;
- If a and b are types, then $\langle a, b\rangle$ is a type;
- If a is a type, then $\langle s, a\rangle$ is a type;
- All types are obtained by applying the induction rules of a finite number of times.

3. Montague's grammar

In his paper "The proper treatment of quantification in ordinary Englis" (1973), Montague formulated a syntax and a semantics for natural language (NL) which has the same rigor and precision as the syntax and semantics of formal languages. He introduced for NL a categorial grammar with the name PTQ, from the name of the paper. This grammar PTQ uses 11 types (syntactic categories) of words: *sentences*, *intransitive verbs, term(en)s (NP, proper nouns, pronouns,etc), transitive verbs, adverbs of type VP, sentence adverbs, common nouns, prepositions, sentence complement verbs, (believe, assert, etc, used with "that"), infinitive complement verbs (try, wish, etc)* , *determiners* (*every, the, a, an*).

In the categorial grammar PTQ of Montague, the type (syntactic categorie) is defined as [7, 8, 3]

- t is a syntactic category, of the expressions to which a truth value can be assigned, i.e. the category of sentences;
- e is a syntactic category, of the expressions to which entities can be assigned;
- if A is a syntactic category, and B is a syntactic category then A/B is a syntactic category.

The *rule of the categorial cancellation* is the following: *if an expression of category A/B combines with an expression of category B then is proceeded an expression of category A.*

The nine categories above have "predefinited" types: for example, intransitive verbs are of type t/e, terms have the type $t/(t/e)$, "believes that" have the type $(t/e)/t$, etc (as in the bellow figure).

If we would analyses how the sentences *John walks* or *John believes that Mary walks*, are obtained by the categorial cancellation rule we shall obtain sentences of type t. Indeed, denoting by + juxtaposition, we have:

$$John_{t/(t,e)} + walks_{t/e} = (John\ walks)_t$$

$$John_{t/(t,e)}+(believes\, that)_{(t/e)/t}+(Mary\, walks)_t = (John\, believes\, that\, Mary\, walks)_t.$$

Definition:

A sentence is any recursive combination of basic expressions that produces, after a finite number of applications of the categorial cancellation rule, an expression of category t.

In Table 1 we will present the syntactic categories of the PTQ grammar.

3.1. **Syntactic rules in the PTQ categorial grammar.** In the following we will use the notation: if A is a syntactic category then B_A is the set of words in the dictionary(lexical entries) of category A and P_A is the set of groups of words of category A. For $A = t$, B_t is the empty set.

The first of the syntactic rules is the following:

Name	Categorial definition	Denotation	Example
t	$-$	$sentence$	$-$
IV	t/e	$VP; intrans.verb$	$run, walk, talk$
$T(term)$	$t/IV or\ t/(t/e)$	$NP; proper\,name$	$John, Mary$
TV	$IV/T or (t/e)/(t/(t/e))$	$transitive\ verb$	$find, eat, love$
IAV	IV/IV	$VP\,adverb$	$rapidly, slowly$
CN	$t//e$	$common\,noun$	$man, woman$
SA	t/t	$sentence\,adverb$	$necessarily$
$Prep$	IAV/T	$preposition$	$in, about$
SCV	IV/t	$sentence\,compl.verb$	$believe, assert$
ICV	$IV//IV$	$infinitive\,compl.verb$	$try, wish$
DET	T/CN	$determiner$	$every, the, a, an$

TABLE 1. The syntactic categories of the PTQ grammar

- S1. For each syntactic category A, the set B_A is included in P_A.
  All the syntactic rules forming complex expressions have the general form:
- Si: If $\alpha \in P_{A/B}$ and if $\beta \in P_B$ then $F_i(\alpha, \beta) \in P_A$.

Some examples of the rules proposed by Montague are given hereafter :

- S2 (Complex expressions of category Term):
  If $\alpha \in P_{T/CN}$ and if $\beta \in P_{CN}$ then $F_2(\alpha, \beta) \in P_T$.
  The function $F_2(\alpha, \beta)$ is $\alpha^*\beta$ where α^* is α except in the case where α is a and the first word in β begins with a vowel; in this case $\alpha^* = an$.
- S4 (Complex expressions of category t, sentences):
  If $\alpha \in P_T$ and if $\beta \in P_{IV}$ then $F_4(\alpha, \beta) \in P_t$.
  The function $F_4(\alpha, \beta)$ is $\alpha\beta^*$ where β^* is the result of replacing the first verb from β by its third person singular present form.

The rules introduced by Montague permit the translation in *de dicto* (non-specific) mode and also in *de re* (specific) mode for a sentence.

He first defined a correspondence between syntactic category and the types in the form of a function f.

Definition:

- $f(t) = t$;
- $f(CN) = f(IV) = \langle e, t\rangle$;
- $f(A/B) = \langle\langle s, f(B)\rangle, f(A)\rangle$ for all syntactic category A and B.

The correspondence between syntactic categories and types is indicated in Table 2.

Beside the types, the expressions themselves are translated. Let us observe first that basic expressions (lexical entries) are translated in constants of type $\langle e, t\rangle$

Syntactic category	Type
t	t
IV	$\langle e, t\rangle$
CN	$\langle e, t\rangle$
$T(term) = t/IV$	$\langle\langle s, \langle e, t\rangle\rangle, t\rangle$
$TV = IV/T$	$\langle\langle s, \langle\langle s, \langle e, t\rangle\rangle, t\rangle\rangle, \langle e, t\rangle\rangle$
$IAV = IV/IV$	$\langle\langle s, \langle e, t\rangle\rangle, \langle e, t\rangle\rangle$
T/CN	$\langle\langle s, \langle e, t\rangle\rangle, \langle\langle s, \langle e, t\rangle\rangle, t\rangle\rangle$
t/t	$\langle\langle s, t\rangle, t\rangle$
$SCV = IV/t$	$\langle\langle s, t\rangle, \langle e, t\rangle\rangle$
$ICV = IV//IV$	$\langle\langle s, \langle e, t\rangle\rangle, \langle e, t\rangle\rangle$
IAV/T	$-$

TABLE 2

Formal expression	Associated type
P	$\langle s, \langle e, t\rangle\rangle$
P^e	$\langle e, t\rangle$
j	e
$P^e(j)$	$t(categorial\ cancellation\ rule)$
$\lambda P[P^e(j)]$	$\langle\langle s, \langle e, t\rangle\rangle, t\rangle(\ cancellation\ rule\ for \lambda expressions)$

TABLE 3

as for example: lexical entries *man* and *walk* are translated in *man'* and *walk'*. The translation of proper nouns is defined as follows: *John* has as translation $John' = \lambda P[P^e(j)]$ where P is a predicate variable and j is a constant which represents *John*. Let us verify that $\lambda P[P^e(j)]$ is of type specific for terms, that means $\langle\langle s, \langle e, t\rangle\rangle, t\rangle$. See Table 3.

For translation formalism is enough to define for each syntactic functional application rule Sj (j=1,...14) of the form:

Sj: If $\alpha \in P_{A/B}$ and if $\beta \in P_B$ then $F_j(\alpha, \beta) \in P_A$

a translation rule Tj of the form:

Tj: If $\alpha \in P_{A/B}$ and $\beta \in P_B$, and if α and β translate into α' and β' respectively, then $F_j(\alpha, \beta)$ translates into $\alpha'(\beta'^i)$.

Example:

Let us compute translation of *Mary talks*. In grammar PTQ we have:

$Mary \in B_T$ so, by S1, $Mary \in P_T$.

$talk \in B_{IV}$ and, by S1, $talk \in P_{IV}$.

By S4, $Mary\, talks \in P_t$.

The translation of $Mary$ is $\lambda P[P^e(m)]$, and the translation of $talk$ este $talk'$. Henceforth, the translation of $F_4(Mary, talk)$ is $\alpha'(\beta'^i) = \lambda P[P^e(m)](talk'^i)$. This last formula may be simplified to

$\lambda P[P^e(m)](talk'^i) = talk'^{e,i}(m) = talk'(m)$.

The first simplification is a $\lambda\, conversion$, the second is by the rule ^{i,e}.

Example

We can verify that the determiner *every* (an expression of the category T/CN) has the appropriate type:

$f(T/CN) = f((t/IV)/CN) = \langle\langle s, f(CN)\rangle, f(t/IV)\rangle = \langle\langle s, \langle e,t\rangle\rangle, \langle\langle s, f(IV)\rangle, f(t)\rangle\rangle = \langle\langle s, \rangle e, t\rangle\rangle, \langle\langle s, \langle e,t\rangle\rangle, t\rangle\rangle$.

Indeed, for *every*, which is translated in $\lambda Z[\lambda Y[\forall x(Z^e(x) \to Y^e(x))]]$ we have: expression $\forall x(Z^e(x) \to Y^e(x))$ is of type t. As Y is of type $\langle s, \langle e,t\rangle\rangle$, expression $\lambda Y[\forall x(Z^e(x) \to Y^e(x))$ is of type $\langle\langle s, \langle e,t\rangle\rangle, t\rangle$. Z is also of type $\langle s, \langle e,t\rangle\rangle$, and then $\lambda Z[\lambda Y[\forall x(Z^e(x) \to Y^e(x))]]$ is of $\langle\langle s, \langle e,t\rangle\rangle, \langle\langle s, \langle e,t\rangle\rangle, t\rangle\rangle$.

4. The treatment of quantitative sentences

In the following we will try to explain the utility of intensional logic for the semantic representation of some significant quantitative sentences. This kind of natural language sentences presents a special importance because of it's mostly use as query language over the internet and most of these sentences refer to quantity (products, money). On the other hand these types of sentences can also be used for the acquisition of new knowledge in a knowledge base which has as input natural language sentences.

For an easier exemplification we split the quantitative sentences in three categories:

• Definite quantity sentences (those sentences of which quantifiers represent exactly the expressed quantity). Example: *Four men cry. John eats an apple.*

• Indefinite quantity sentences (the quantifiers of these sentences gives us an approximation of the expressed quantity without specify it). Example: *Most women cry. A number of people read.*

• Restrictive quantity sentences (in this case the quantifiers restrict with precision the expressed quantity). Example: *Maximum five children answer.*

Generally speaking, the quantitative sentences are generated by the presence of numerals, but there are also cases when there aren't any numeral in these sentences (for example, the indefinite quantity sentences). This is why we will first try to present the way that these numerals are translated in the intensional logic.

There are two types of numerals with more importance in the construction of quantitative sentences, numerals which will be translated in different way:

a) Singles numerals which are determinants of the sentences where they belong (e.g. *Five man laugh*).

b) Numerals which are preceded by an adverb with the role of pre-determinant, with which it forms the sentence determinant. (e.g. *Maximum eleven people left the room*).

In the first case the numerals represent the sentence determinant, so their type is T/CN and their translation in the intensional logic will be:

$\lambda Z[\lambda Y[\exists_N X(Z^e(X) \wedge Y^e(X) \wedge N = j))]]$ - where j represents the number indicated by the numeral

In the construction of this expression we used the notation given in [3]. In the following we will prove that this expression is of type T/CN:

– Z^e is of type $\langle e,t\rangle$ and X is of type e $\Rightarrow$ (with the use of cancellation rule) $Z^e(X)$ is of type t;

– in the same way we can prove that the expression $Y^e(X)$ is of type t;

– N and j are of the same type e $\Rightarrow$ (using the definition of the intensional logic) the expression N=j is of type t;

– again using the definition of intensional logic which says that: if two expressions A and B have the same type t, then the expression A$\wedge$B is also of type t. Using this definition on the above expression $\Rightarrow$ the expression $Z^e(X) \wedge Y^e(X) \wedge N = j$ is of type t;

– if the expression $Z^e(X) \wedge Y^e(X) \wedge N = j$ is of type t $\Rightarrow$ the expression $\exists_N X(Z^e(X) \wedge Y^e(X) \wedge N = j)$ is also of type t;

– the expression Y is of type $\langle s,\langle e,t\rangle\rangle \Rightarrow \lambda Y[\exists_N X(Z^e(X) \wedge Y^e(X) \wedge N = j)]$ has the type $\langle\langle s,\langle e,t\rangle\rangle,t\rangle$;

– the expression Z is of type $\langle s,\langle e,t\rangle\rangle \Rightarrow \lambda Z[\lambda Y[\exists_N X(Z^e(X) \wedge Y^e(X) \wedge N = j)]]$ has the type $\langle\langle s,\langle e,t\rangle\rangle,\langle\langle s,\langle e,t\rangle\rangle,t\rangle$, type which is T/CN $\square$

In the second situation the numerals will be translated such as base expression by their semantics (value for the numeral). For example, the numeral *four* has the translation 4'.

In the following we will explain (based on an example) the way that definite quantity sentences will be translated in the intensional logic form (in this example, we will also show the way that differed type of ambiguities are solved and how we represent the **a** and the **the** determinants). Let us consider the following natural language sentence that we have to translate in its corresponding intensional logic formula:

Five men enter a room.

First of all we observe that the sentence is composed by the following atoms:

five – numeral with determinant role

men – common noun;

enter – transitive verb;

a – determinant;

room – common noun;

FIGURE 1. The derivation tree for the sentence: *Five men enter a room*

First we prove that this sentence is semantically correct (which is equivalent to prove that the sentence type is t). Next we will show that we can map this sentence in an intensional logic formula.

To prove that the type of this sentence is t, we used a bottom-up algorithm which constructs the sentence from its atomic parts and then, using the cancellation rule, groups these atoms (see Figure 1).

It can be notice that this is not the only derivation three which we can construct (at the second step of the construction we could apply the cancellation rule to couple enter with *five men* instead with the sequence *a man* - this is named the de re interpretation). This two way interpretation of the sentence is caused by its ambiguity introduced by the scope of the quantifiers.

Now we try to translate this sentence. As we saw earlier, there are two interpretations for this sentence: de dicto and de re. Each type of interpretation will gives us another formula of the intensional logic (this is possible because of the semantic ambiguity introduced by this sentence).

To translate this sentence we have to add a new cancellation rule for the composition of the sequence α of any type and the sequence β of type T. Thus, the cancellation rule for the sequence $\alpha\beta$ is (F_T) $F_T(\alpha,\beta) = \beta'(\alpha'^i)$ where α', β' represents the translation of the α, β respectively.

First, we will analyze the de dicto interpretation for the sentence translation. For the sentence translation we have to follow the steps below:

• *five*$\in P_{T/CN}$ and this expression translation is (with the respect of what we had shown at the beginning of this chapter) $\lambda Z[\lambda Y[\exists_N X(Z^e(X) \wedge Y^e(X) \wedge N = 5)]]$;

• *men*$\in P_{CN}$ and its translation is *men*'. Thus, the expression *five men* will be translated (after the use of the λ-conversion and the ^{i,e}-rule) as:

$\lambda Y[\exists_N X(men'(X) \wedge Y^e(X) \wedge N = 5)]]$ (which has the type T);

• *a*$\in P_{T/CN}$ and the translation of this expression is $\lambda A[\lambda B[\exists_1 J(A^e(J) \wedge B^e(J))]]$ (to be more specific, we use another set of variables - the notation is taken from [4]); *room* $\in P_{CN}$ and its translation is *room*'. Thus, after applying the cancellation rule and after the use of the λ-conversion and the ^{i,e}-rule, the expression *a room* will be $\lambda B[\in_1 J(room'(J) \wedge B^e(J))]$ whose type is T.

• *enter*$\in P_{IV/T}$ with the translation $enter'^2$ (where the numeral 2 means that this predicate needs two arguments - this can be also expressed with the help of lambda calculus:$\lambda A[\lambda B(enter(A,B))]$). After the use of the cancellation rule to the expressions *enter* (of type IV/T) and *a room* (of type T) we receive (we use the F_T cancellation rule because the type of the expression *a room* is T) - after applying the λ-conversion and the ^{i,e}-rule - the expression $\exists_1 J(room'(J) \wedge enter'^1(J))$ whose type is IV (the predicate $enter'^1$ means that the predicate *enter*' needs another argument)

• Now we only have to couple the remaining two expression: *five men* (whose type is T) and *enter a room* (whose type is IV), which has the following translations $\lambda Y[\exists_N X(men'(X) \wedge Y^e(X) \wedge N = 5)]]$ and $\exists_1 J(room'(J) \wedge enter'^1(J))$ respectively. After applying the λ-conversion and after applying the ^{i,e}-rule we will obtain the following expression of the intensional logic:

$$\exists_N X(men'(X) \wedge \exists_1 J(room'(J) \wedge enter'^1(J))(X) \wedge N = 5) \Longleftrightarrow$$

$$\exists_N X(men'(X) \wedge (\exists_1 J(room'(J) \wedge enter'(J,X))) \wedge N = 5)$$

The semantic of this formula express the fact that there are exactly *five men* who enter in a room (not necessarily the same room).

To obtain the other translation of the sentence we must follow the de re interpretation (we will give only the important steps of the process - the steps which was excluded are the same with the previous interpretation):

• *five* $\in P_{T/CN}$ and this expression translation is (with the respect of what we had shown at the beginning of this chapter) $\lambda Z[\lambda Y[\exists_N X(Z^e(X) \wedge Y^e(X) \wedge N = 5)]]$;

• *men* $\in P_{CN}$ and its translation is *men*'. Thus, the expression *five men* will be translated (after applying the λ-conversion and the ^{i,e}-rule) as:

$$\lambda Y[\lambda_N X(men'(X) \wedge Y^e(X) \wedge N = 5)]]$$

(which has the type T);

• *enter* $\in P_{IV/T}$ with the translation $enter'^2$. This, the expression *Five men enter* will be mapped in:

$$\exists_N X(men'(X) \wedge enter'^1(X) \wedge N = 5)]$$

(whose type is t/T); (I)

• $a \in P_{T/CN}$ and the translation of this expression is $\lambda A[\lambda B[\exists_1 J(A^e(J) \wedge B^e(J))]]$ (to be more specific, we use another set of variables - the notation is taken from [4]); *room* $\in P_{CN}$ and its translation is *room*'. Thus, after applying the cancellation rule and after the use of the λ-conversion and the i,e-rule, the expression a room will be $\lambda B[\exists_1 J(room'(J) \wedge B^e(J))]$whose type is T; (II)

• after the combining the two expressions (I) and (II) and after using the cancellation rule F_T we will obtain the following formula:

$$\exists_1 J(room'(J) \wedge (\exists_N X(men'(X) \wedge enter'^1(X) \wedge N = 5))(J)) \Longleftrightarrow$$

$$\exists_1 J(room'(J) \wedge (\exists_N X(men'(X) \wedge enter'(X, J) \wedge N = 5)))$$

The semantic of this formula express the fact that there is exactly one room in which five men enter. In the same way we can translate the sentence *Five men enter the room* (formula which is obtained by replacing the "a" determinant with the "the" determinant) which has the following to translation (corresponding to the de dicto and de re interpretation):

$$\exists_N X(men'(X) \wedge (\exists_1 J(\forall K(room'(J) \equiv J = K) \wedge enter'(J, X))) \wedge N = 5)$$

(de re) and

$$\exists_1 J((\forall K(room'(J) \equiv J = K) \wedge (\exists_N X(men'(X) \wedge enter'(X, J) \wedge N = 5)))$$

(de dicto)

As it may be seen these two expressions are equivalent semantically speaking.

Now we show how we can map another type of quantitative sentences: the restrictive quantity sentences. As we had done in the previous case we will start to analyze an example sentence. Thus, let consider the sentence: *Minimum ten men laugh*, where:

minimum – adverb (pre-determinant) with the type $T/CN/T/CN$;

ten – numeral;

men – noun;

laugh – intransitive verb.

It can be prove that this sentence is correct (from the syntactic point of view) in a similar way as it was proved for the exact quantity sentence.

For the formula translation we have to follow the following steps (as we had done earlier with the other example):

• *minimum* $\in P_{T/CN/T/CN}$ and its translation:

$$\lambda K[\lambda Z[\lambda Y[\exists_N X(Z^e(X) \wedge Y^e(X) \wedge minimum'(N, K^e))]]]$$

Which has the type T/CN/T/CN (to prove the type we must do the steps shown for the numeral)

• *ten* $\in P_{T/CN}$ and its translation will be *10*'. After applying the cancellation rule for the above two expression (after applying the λ-conversion and the ^{i,e}-rule) we obtain:

$$\lambda Z[\lambda Y[\exists_N X(Z^e(X) \wedge Y^e(X) \wedge minimum'(N, 10'))]];$$

• *men* $\in P_{CN}$ which translation is *men*'. Thus, the translation of the expression *minimum ten men* (after applying the λ-conversion and the ^{i,e}-rule) is:

$$\lambda Y[\exists N X(men'(X) \wedge Y^e(X) \wedge minimum'(N, 10'))];$$

• *laugh* $\in P_{IV}$ with the translation *laugh*'. Thus the final translation of the sentence minimum ten men laugh (after applying the λ-conversion and the ^{i,e}-rule) will be:

$$\exists_N X(men'(X) \wedge laugh'(X) \wedge minimum'(N, 10'))$$

- with the type t;

The semantic of this formula means that there are minimum ten men who laugh. It must be remarked that in this case we had also a λ-variable (K) which took place as a predicate argument (predicate which is also expressed by an λ-variable).

The last type of sentences which will be discussed here is the indefinite quantity sentences. This type of sentences does not need the presence of the numeral. For this kind of sentences we choose the following sentence: *Most people run* where the syntactic types of its atoms are:

most – determinant;

people – noun

run – intransitive verb

Similarly to the previous two formulas we can prove that this formula type is t (this, the sentence is correct in the syntactic point of view).

As we have already seen in the other cases, we will follow the next steps:

• *most* $\in P_{T/CN}$ with the translation:

$$\lambda Z[\lambda Y[\exists_N X(Z^e(X) \wedge Y^e(X) \wedge most'(N))]]$$

- of the type T/CN

• *people* $\in P_{CN}$ its translation is *people*'. After applying the cancellation rule with the expression most (after applying the λ-conversion and the ^{i,e}-rule) we obtain:

$$\lambda Y[\exists_N X(people(X) \wedge Y^e(X) \wedge most'(N))]$$

- of type T

• *run* $\in P_{IV}$ with the translation *run*'. After applying the cancellation rule with the above expression (after applying the λ-conversion and the ^{i,e}-rule) we obtain the final formula:$\exists_N X(people(X) \wedge run'(X) \wedge most'(N))$ - which expresses the fact that there are N people who run and N follows the **most**' predicate.

By the previous three examples we wanted to show the importance of the categorical grammars and of the intensional logic in the natural language representation in general and of the quantitative sentences in particular. As we have seen, this translation technique of the natural language semantic can be used successfully in real life. It can also be applied to express more complex sentences that can hardly solved with the use of other grammars.

References

[1] J.Allen : "Natural language understanding", Benjamin/Cummings Publ., 2nd ed., 1995.

[2] D. Jurafsky, J. Martin: "Speech and language processing", Prentice Hall, 2000.

[3] G. Morrill:"Type Logical Grammar.Categorial Logic of Signs", Kluwer Academic Publishers, 1994.

[4] A. Onet, D. Tatar: "Semantic representation of the quantitative natural languange sentences", Studia Univ. "Babes-Bolyai", Seria Informatica, 1999, nr 2, pg 99-109.

[5] A. Onet, D. Tatar: "Semantic analysis in dialogue interfaces", Studia Univ. "Babes-Bolyai", Seria Informatica, 2000, nr 1, pp79-89.

[6] D. Tatar: "Inteligenta artificiala: demonstrare automata de teoreme, prelucrarea limbajului natural", Editura Microinformatica, 2001.

[7] A. Thayse (editor): "From modal logic to deductive databases", John Wiley and Sons, 1989.

[8] A. Thayse (editor): "From natural language processing to logic for expert systems", John Wiley and Sons, 1990.

Department of Computer Science, Faculty of Mathematics and Computer Science, "Babeş-Bolyai" University, Cluj-Napoca, Romania

E-mail address: adrian|dtatar@cs.ubbcluj.ro