COLING 82, J. Horecký (ed.)
North-Holland Publishing Company
© Academia, 1982

A LESNIEWSKIAN VERSION OF MONTAGUE GRAMMAR

Arata ISHIMOTO

Department of Industrial Administration
Science University of Tokyo
Noda, Chiba
Japan

We shall be concerned in this paper with the logical analysis of natural language on the basis of Lesniewski´s ontology, which is a logical system without type-distinction between individuals and monadic predicates. This, it is believed, is also one of the features of natural language, and use will be made of this feature for developing a fragment of natural language.

1 Introduction According to Montague every simple sentence of natural language is of the form

$$\mathrm{Np} + \mathrm{Vp} = (\mathrm{Det} + \mathrm{N}) + \mathrm{Vp},$$

and, as emphasized by Barwise-Cooper [1] Det + N constitutes a generalized quantifier to be applied to Vp or verb phrase as one-place predicate. (For Montague grammar consult among others Montague[10], Cresswell [3] and Jirku [6].)

Thus, simple sentences such as:

1.1 the man walks,
1.2 every man walks,
1.3 some (a) men (man) walk(s)
1.4 (at least) two men walk,
1.5 (at least) three men walk,

are respectively of the following logical forms:

1.11 $(\lambda P \lambda Q P \iota x \text{ man})$ walk,
1.21 $(\lambda P \lambda Q \forall x (P\ x \supset Q\ x) \text{ man})$ walk,
1.31 $(\lambda P \lambda Q \exists x (P\ x \wedge Q\ x) \text{ man})$ walk,
1.41 $(\lambda P \lambda Q \exists x \exists y (P\ x \wedge P\ y \wedge P\ x \wedge Q\ y \wedge x \neq y) \text{ man})$ walk,
1.51 $(\lambda P \lambda Q \exists x \exists y \exists z (P\ x \wedge Q\ y \wedge Q\ x \wedge Q\ y \wedge P\ z \wedge Q\ z \wedge$
$x \neq y \wedge y \neq z \wedge z \neq x) \text{ man})$ walk,

where $Q \iota x P\ x$ is the Russellian-type definite description to be defined as

$$\exists x (P\ x \wedge Q\ x) \wedge \forall x \forall y (P\ x \wedge P\ y . \supset x = y),$$

with the scope restricted to Q.

By means of λ-conversion, for example, from 1.11 and 1.21 we respectively obtain:

$$\text{walk } \iota x \text{ man } x,$$

and

$$\forall x (\text{man } x \supset \text{walk } x),$$

so that Det + N now becomes in both cases a generalized quantifier.

(Here as well as in what follows intension will not be taken into consideration. The proposed Lesniewskian theory will therefore be extensional.)

Now it is to be remembered that 1.11-1.51 are the formulas of (second-order) predicate logic. In other words, the sentences of natural language are embedded in predicate logic although a large number of formulas of the logic do not have their counterparts in natural language.

However, the embedding in traditional predicate logic is not necessarily the only possible way for us to understand the logical structure of natural language. In fact, natural language could also be embedded in Lesniewski´s ontology augmented by a number of additional notions so that a fragment of natural language can be accomodated there, and in what follows we shall be concerned with a detailed construction of Lesniewskian-type logical grammar.

2 Logical grammar based upon Lesniewski's ontology In Lesniewskian-type Montague grammar we are all the same starting from the structural assumption of the simple sentences of natural language as mentioned at the beginning of the last section. Nevertheless, in Lesniewskian version of logical grammar, which will be abbreviated as LMG in the sequel, Det, i.e. determiners, are represented not by generalized quantifiers but by the functors of Lesniewski´s ontology with noun and intransitive verb phrases to be combined thereby as two arguments, which are now provided with the category of names in the sense of Lesniewski's ontology, not the one corresponding to monadic predicates of predicate logic.

Without going into the details of the phrase structure and transformational rules necessary for generating a fragment of English (which is by far smaller than that proposed by Cooper-Parsons [2]) we shall present the deep or logical structures of a number of simple sentences (of English) as (well-formed) sentences belonging to the proposed Lesniewskian version of Montague grammar to be designated as LMG:

2.11 ((the man) walk),
2.12 ((every ((man or) woman)) speak),
2.13 ((some woman) not play)),
2.14 ((every man)[love (some woman)]$_1$),
2.15 (the woman)[(love and) admire)
(every (boy (or girl)]$_1$)
2.16 (not ((every man) speak)),

where [ ]$_1$ is a combinator or operator which makes an intransitive verb phrase out of a transitive verb and a noun phrase. This combinator is represented by a declension in inflectional languages such as Slavic ones, while in the case of uninflected languages such as English it is taken care of by word order.

It is remarked that any combinator could sometimes be applied from the left to the right as suggested by Cresswell [3]. This has already been practised in some of the above sample sentences. Thus such sentences like:

((every man) [(some (dog (or cat)) love]$_1$),

and

((some woman) [(some man)(love not)]$_1$),

are also well-formed, being close to the word order usual in Japanese.

As is well known, 2.14 has another deep or logical structure in quantificational theory with 'some' having the wider scope than 'every'. In this case, the given sentence is of the form:

([(every man) love]$_2$ (some woman)),

where [ ]$_2$ combines a noun phrase and a transitive verb phrase giving rise to an expression corresponding to a monadic predicate. But, unlike [ ]$_1$, the noun phrase to be combined is in the nominative case.

The use of [ ]$_2$ will be illustrated as follows:

2.21 ([(some man) hate]$_2$ (every woman)),

2.22 ((every woman) [(some man) hate]$_2$),

which are of the same structure with each other with 'some man' remaining the subject of these sentences.

As is easily understood from the development up to the present, the (well-formed) expressions of the proposed LMG as a logic are defined in terms of the expressions both constant and variable having the category of names in the sense of Lesniewski's ontology and relations as well as of a number of logical operators not only sentential and quantificational but also name-forming and relation-forming. (For Lesniewski's ontology consult Iwanus [5], Luschei [9] and Slupecki [11].)

3 Axiomatization of LMG as a logic If we are to develop LMG as a logic,we have to axiomatize it as a logical system.

Fortunately the axiomatization of Lesniewski's ontology has been intensively worked out ever since its single axiom was first proposed by Lesniewski himself in 1921.

Thus, we are starting with the celebrated single axiom originating from Lesniewski:

$$3.01 \quad \begin{aligned} \varepsilon(a, b) \equiv .&(\exists x)\varepsilon(x, a) \\ &\wedge(x)(\varepsilon(x, a) \supset \varepsilon(x, b)) \\ &\wedge(x)(y)(\varepsilon(x, a) \wedge \varepsilon(y, a) . \supset \varepsilon(x, y)), \end{aligned}$$

or its simplified version by Sobocinski [12] :

$$3.02 \quad \begin{aligned} \varepsilon(a, b) \equiv &(\exists x)(\varepsilon(x, a) \wedge \varepsilon(x, b)) \\ &(x)(y)(\varepsilon(x, a) \wedge \varepsilon(y, a) . \supset \varepsilon(x, y)), \end{aligned}$$

where ε stands for 'the' and a, b,... and the like are (meta-) logical variables ranging over the expressions of the category of Lesniewskian names. On the other hand, $\varepsilon(a, b)$ stands for $((\varepsilon a)\ b)$ or $((a\ \varepsilon)\ b)$ or $(b\ (\varepsilon a))$ or $(b\ (a\ \varepsilon))$, which are forthcoming as a result of the liberalization due to Cresswell [3]. Analogously, A(a, b) (I(a, b)) represents ((A a) b) etc. ((I a) b) etc.) with A (I) taking the place of 'every' ('some'). (A and I are also known as syllogistic functors corresponding to 'every' and 'some' respectively.)

Nevertheless, 3.01 or 3.02 is not enough to develop LMG as a language. In fact, we need a number of additional axiom (schemata) for taking care of name- and relation-forming (logical) operators and the expressions involving [ ]$_1$ and [ ]$_2$.

The axiom (schemata) stipulating these operators are well-known, being of the forms:

3.11 $\varepsilon(a, b \text{ and } c) \equiv .\varepsilon(a, b) \wedge \varepsilon(a, c)$,

3.12 $\varepsilon(a, b \text{ or } c) \equiv .\varepsilon(a, b) \vee \varepsilon(a, c)$,

3.21 $\varepsilon(a, \text{not } b) \equiv .\varepsilon(a, a) \wedge \sim\varepsilon(a, b)$,

3.22 $(R \text{ and } S)(a, b) \equiv .R(a, b) \wedge S(a, b)$,

3.23 $(R \text{ or } S)(a, b) \equiv .R(a, b) \vee S(a, b)$,

3.24 $\text{not } R(a, b) \equiv .\varepsilon(a, a) \wedge \varepsilon(b, b) \wedge \sim R(a, b)$,

3.25 $\text{en } R(a, b) \equiv R(b, a)$

3.26 $R(a, b) \supset .\varepsilon(a, a) \wedge \varepsilon(b, b)$,

where in 3.26 R is atomic. We are also abbreviating such expressions as ((R and)S)and the like as (R and S) for the purpose of perspicuity.

Lemma 3.3 3.25 holds of any relation R.

This is easily proved on the basis of 3.23-3.24 by induction on the length of the given R.

$$3.41 \quad A(a,\ b) \equiv (x)(\ \varepsilon(x,\ a) \supset \varepsilon(x,\ b),$$
$$3.42 \quad I(a,\ b) \equiv \exists x)(\ \varepsilon(x,\ a) \wedge \varepsilon(x,\ b)).$$

We are now presenting some of the axiom (schemata), which take care of the expressions containing $[\ ]_1$ and $[\ ]_2$.

$$\begin{aligned}
3.51 \quad & \varepsilon(a,\ [(\varepsilon\ b)\ R]_1 \\
& \equiv .(\exists x)(\ \varepsilon(x,\ a) \wedge (\exists z)(\ \varepsilon(z,\ b) \wedge R(x,\ z)) \\
& \quad \wedge(x)(y)(\ \varepsilon(x,\ a) \wedge \varepsilon(y,\ a).\supset \varepsilon(x,\ y)) \\
& \quad \wedge(x)(y)(\ \varepsilon(x,\ b) \wedge \varepsilon(y,\ b).\supset \varepsilon(x,\ Y) \\
3.52 \quad & A(a,\ [(I\ b)\ R]_1 \\
& \equiv (x)(\ \varepsilon(x,\ a) \supset (\exists y)(\ \varepsilon(y,\ b) \wedge R(x,\ y)) \\
3.53 \quad & I(b,\ [(A\ a)\ R]_2\ ) \\
& \equiv (\exists y)(\ \varepsilon(y,\ b) \wedge (x)(\ \varepsilon(x,\ a) \supset R(x,\ y))
\end{aligned}$$

It is noticed that some of these axioms are not well-formed as sentences of LMG as a language although they belong to LMG qua logic.

Theorem 3.7 Every simple sentence of LMG (as a language) is equivalent to a sentence (of LMG as a logic), and this sentence involve only ε and atomic relations besides logical operators with quantifiers binding only such name variables x and y as occur there in the context ε(x, a) or R(x, y).

The proof is carried out by induction on the number of symbols other than those mentioned in the theorem on the basis of axioms.

It is again observed that the formulas to which these sentences of natural language are transformed are not necessarily those belonging to LMG as a language.

4 Translation of LMG into predicate logic

It will be shown in this section that LMG as a language is embedded in first-order predicate logic (with equality) via a translation T to be defined presently. (The proposed translation dates from Prior [8] and has been elaborated by Ishimoto [4] and Kobayashi-Ishimoto [7].)

The translation T is defined by induction on the number of the words contained in the given expression of LMG.

In the first place, the basis is taken care of by:

$$\begin{aligned}
& T\ a = F_a, \\
& T\ b = F_b, \\
& \quad \ldots\ ,
\end{aligned}$$

where a, b,... are (atomic) names constant and variable, and F_a, F_b, ... are monadic predicates again constant or variable corresponding to a, b,... not necessarily exhausting all of them.

$$\begin{aligned}
& T\ \varepsilon = \lambda P \lambda Q Q \iota x P\ x, \\
& T\ A = \lambda P \lambda Q \forall x(P\ x \supset Q\ x), \\
& T\ I = \lambda P \lambda Q \exists x(P\ x \wedge Q\ x).
\end{aligned}$$

We are now proceeding to the induction steps:

$$\begin{aligned}
& T\ \alpha\wedge\beta = T\ \alpha \wedge T\ \beta, \\
& T\ \alpha\vee\beta = T\ \alpha \vee T\ \beta, \\
& T\ \alpha\supset\beta = T\ \alpha \supset T\ \beta, \\
& T\ {\sim}\alpha \ = {\sim}T\ \alpha, \\
& T\ (x)\alpha = \forall F_x\ T\ \alpha, \\
& T\ (\exists x)\alpha = \exists F_x\ T\ \alpha,
\end{aligned}$$

where α, β,... are meta-logical variables ranging over the sentences of LMG..

Before taking up the translation of relations, i.e. transitive verbs we have to introduce in advance another translation T_1 which transforms every relation in LMG into a (binary) relation of predicate logic. The translation T_1 is defined inductively on the number of the relation-forming operators employed for defining the given one.

Starting with the basis:

$$T_1\ R = \lambda x \lambda y G_R(x, y),$$

where G_R is the (binary) relation (of the predicate logic) corresponding to the give atomic relation of LMG, induction steps are:

$$T_1\ R \text{ and } S = \lambda x \lambda y(T_1\ R(x, y) \wedge T_1\ S(x, y)),$$
$$T_1\ R \text{ or } S = \lambda x \lambda y(T_1\ R(x, y) \vee T_1\ S(x, y)),$$
$$T_1 \text{ not } R = \lambda x \lambda y {\sim} T_1\ R(x, y),$$
$$T_1 \text{ en } R = \lambda x \lambda y T_1\ R(y, x)$$

(T_1 R and S, etc. will be abbreviated as $G_{R \text{ and } S}$,etc.)

On the basis of the translation T_1 thus introduced, T is defined for any relation R of LMG as follows:

$$T\ R = \lambda P \lambda Q T_1\ R(\ \iota x P\ x,\ \iota x Q\ x),$$

with $R(\ \iota x P\ x,\ \iota x Q\ x)$ being defined as:

$$\exists x \exists y(P\ x \wedge Q\ y \wedge R(x, y)) \wedge \forall x \forall y(P\ x \wedge P\ y. \supset x = y)$$
$$\wedge \forall x \forall y(Q\ x \wedge Q\ y. \supset x = y).$$

Lastly the translation T is applied to the operators $[\ ]_1$ and $[\ ]_2$in the following way:

$$T\ [\ ]_1 = \lambda V \lambda W \lambda x(V \lambda y W(x, y)),$$
$$T\ [\ ]_2 = \lambda V \lambda W \lambda y(V \lambda x W(x, y)),$$

where V and W are respectively the variables of the type of noun phrases and (binary) relations in predicate logic.

Availing ourselves of the translation T thus defined, some sample sentences (of LMG) will be translated into the corresponding sentences of predicate logic.

4.11 $T\ 2.11 = T\ ((\text{the man})\ \text{walk}) = T\ ((\varepsilon\ \text{man})\ \text{walk})$
$= ((T\ \varepsilon\ T\ \text{man})\ T\ \text{walk})$
$= (\lambda P \lambda Q Q \iota x P\ x)\ F_{man})\ F_{walk} = F_{walk} \iota x F_{man} x,$

4.14 $T\ 2.14 = T\ ((\text{every man})\ [\text{love (some woman)}]_1)$
$= (\lambda P \lambda Q \forall x(P\ x \supset Q\ x)\ F_{man})$
$((\lambda V \lambda W \lambda x(V \lambda y W(x, y)))(\lambda P \lambda Q \exists x(P\ x \wedge Q\ x)\ F_{woman})\ G_{love})$
$= \lambda Q \forall x(F_{man} x \supset Q\ x) \lambda x(\lambda Q \exists y(F_{woman} y \wedge Q\ y)\ \lambda y G_{love}(x, y))$
$= \forall x(F_{man} x \supset \exists y(F_{woman} y \wedge G_{love}(x, y))).$

As has been exemplified by the above translations we easily obtain:

Lemma 4.3 Every sentence of LMG as a language is translated by T into a formula of first-order predicate logic with equality.

Lemma 4.4 The translation of the theses of LMG as a logic are provable in predicate logic.

The proof is carried out by induction on the length of the proof.

The treatment of the basis will be illustrated by the following example:

4.41 $T\ 3.02 \equiv :T\ \varepsilon(a, b) \equiv .(\exists x)(\ \varepsilon(x, a) \wedge \varepsilon(x, b))$
$(x)(y)(\ \varepsilon(x, a) \wedge \varepsilon(x, b). \supset \varepsilon(x, y)):$
$\equiv :F_b \iota x F_a x \equiv .\ \exists F_x(F_a \iota x F_x x \wedge F_b \iota x F_x x) \wedge$
$\forall F_x \forall F_y(F_a \iota x F_x x \wedge F_a \iota x F_y x. \supset F_y \iota x F_x x):$
$\equiv :F_b \iota x F_a \equiv .\ \exists x(F_a x \wedge F_b x) \wedge \forall x \forall y(F_a x \wedge F_a y. \supset x = y).$

Here use is made of some theses of second-order predicate logic.

All the other axioms, if translated by T, will turn our to be provable in (higher-order) predicate logic. The induction steps do not present any difficulties.

In view of Lemmas 4.3 and 4.4 we obtain,

Corollary 4.5 If a sentence of LMG as language is a thesis of LMG as a logic, then its T-transform is provable in first-order predicate logic with equality.

It is remarked that the proof of the T-transform of a sentence belonging to LMG as a language might involve formulas not necessarily belonging to first-order predicate logic.

Lastly we wish to state without proof a lemma of fundamental importance, namely,

Lemma 4.6 If the T-transform of a sentence belonging to LMG as a language is provable in first-order predicate logic with equality, then the sentence is a thesis of LMG as a logic.

This is proved syntactically as well as semantically by the method employed in Ishimoto [4].

Combining Corollary 4.5 and Lemma 4.6 we obtain,

Theorem 4.7 For every sentence α of LMG as a language $\vdash_{LMG}\alpha$ iff T α is a thesis of first-order predicate logic with equality.

In view of theorem 4.7 as far as the logical derivability of some sentences of natural language as specified above is concerned, there is no difference between first-order predicate logic and Lesniewski's ontology. Use will be made , it is hoped, of this fact in the various fields related to the logical analysis of natural language.

*An earlier version of this paper was read before the Australasian Logic Conference, Wellington, 1981.

References

1 Barwise, J. and Cooper, R., Generalized Quantifiers and Natural Language, Linguistics and Philosophy, 4, 1981, pp. 159-219.

2 Cooper, R. and Parsons, T., Montague Grammar, Generative Semantics and Interpretive Semantics, in Montauge Grammar, ed. by B.H. Partee, New York-San Francisco-London, 1976.

3 Cresswell, M. J., Logics and Languages, London, 1973.

4 Ishimoto, A., A Propositional Fragment of Lesniewski's Ontology, Studia Logica, 36, 1977, pp. 285-299.

5 Iwanus, B., On Lesniewski's elementary ontology, Studia Logica, 31, 1973, pp. 73-125.

6 Jirku, P., Towards an integrated theory of formal and natural language, Kybernetika, 11 (2), 1975, pp. 91-100.

7 Kobayashi, M. and Ishimoto, A., A Propositional Fragment of Lesniewski's ontology and its Formulation by the Tableau Method, forthcoming in Studia Logica.

8 Prior, A. N., +Existence in Lesniewski and Russell, Formal systems and Recursive Functions, Amsterdam, 1963.

9 Luschei, E., The Logical Systems of Lesniewski, Amsterdam, 1963.

10 Montague, R., Formal Philosophy, Selected Papers of R. Montague, New Haven and London, 1974.

11 Slupecki, J., St. Lesniewski's Calculus of Names, Studia Logica, 3, 1955, pp. 7-76.

12 Sobocinski, B., Successive Simplifications of the Axioms Systems of Lesniewski's Ontology, Polish Logic 1920-1939, Oxford, 1967.