KYBERNETIKA — VOLUME 26 (1990), NUMBER 4

ON A LOGICAL FORMALIZATION OF NATURAL LANGUAGE

JAROSLAV PEREGRIN

The possibilities and limitations of a formalization of natural language by means of the common logical systems are investigated. The applicability of the traditional Tarskian model theory to natural language is discussed. The direction which the formalizing approach to natural language should take in order to achieve an adequate account of its semantic aspect is indicated.

1. NATURAL LANGUAGE AS A FORMAL SYSTEM

1.1 Formalization: How and Why?

A description of natural language (NL) has been an objective of linguistic theory since Antiquity. In the recent decades the attempts at a systematic description have been injected by the formalizing mode of scientific thinking which has been influencing more or less all branches of science since the beginning of the present century. The formal description of natural language has proven itself to bring fruitful insight into the nature of our linguistic activity, and, moreover, as NL is not a mere "thing among things" in our world, but rather besides this also a universal means of our acquiring of the world, insight into our grasping of the world.[1]

The importance of NL as our universal means of judging has made it also into an objective of the logical theory. And it was logicians, like Frege, Russell, and others, who have given the first formalizations of NL and who thus have initiated the formalizing mode of reasoning which has found its expression in linguistics later

[1] The first complex formal description of the syntactic structure of NL has been given by Chomsky ([9]); his formalism of generative grammar has been then, in the sixties, further elaborated and modified by numerous authors. In the seventies, the purely syntax-oriented Chomskian approach has been overshadowed by prevailingly semantic formalization done by Montague ([20]) and his followers, while in the present decade still other approaches seem to get into the central place.

in Chomskian and post-Chomskian period.[2] The motivation of the effort of logicians in the field of schematization of NL has been the desire to articulate problems in such a way that they could be treated formally. As Frege ([14]) puts it, "the concept-script should then serve above all to offer the most reliable test of validity of the chain of deductions and point out every premise, which tries to slip into it unnoticed" (p. IV).[3]

In this paper we shall examine the usefulness of the existing systems of formal logic for the sake of formalization of semantics of NL. First we shall characterize NL in a very abstract algebraic way and then we shall test what additional restrictions it would have to fulfill if it ought to be identifiable with this or that logical system. In Section 2 we adopt the view of the model theory: we try to define such class of languages for which models of the kind common in logic, i.e. models interpreting expressions as functions and grammatical rules as some plausible operations with functions, are on hand. In Section 3 we discuss the most evident difficulties connected with the identification of NL with the commonly used logical systems. In Section 4 we indicate in what direction the theory of formal semantics should be developed to provide a quite adequate formal explication of the semantic aspect of NL.

1.2 Language as an Algebra

In general, natural language can be viewed upon as a class of primitive expressions (*the lexicon*) and a class of rules combining expressions into complex expressions (*the grammar*). From the algebraic point of view any grammatical rule can clearly be understood as an n-ary function over the class of language expressions. This class of expressions then splits itself into pairwise disjoint *grammatical categories* in such a way that the domain of any grammatical rule is a Cartesian product of grammatical categories and its range is included in a grammatical category. From this point of view language can be considered as what can be in the most appropriate way called a many-sorted algebra.[4]

Definition 1. A *many-sorted algebra* (MSA) is an ordered pair $\boldsymbol{A} = \langle C, F\rangle$, where (i) $\boldsymbol{C} = \langle C_i\rangle_{i\in I}$ is a family of sets (the *sorts* of $\boldsymbol{A}$); and (ii) $F = \langle F_j\rangle_{j\in J}$ is a family of functions (the *operations* of $\boldsymbol{A}$) such that the domain of each of them is a Cartesian product of sorts and its range is included in a sort. The union of all sorts of $\boldsymbol{A}$ will be called the *carrier* of $\boldsymbol{A}$.

Now let us define several basic notions concerning MSAs.

[2] The influence of Freegean logic on linguistics is thus twofold: it provides both a tool for linguistic analyses and an exemplar of a result of such analysis. This is connected with the twofold role of language in linguistics: its role as a tool and as an objective.

[3] Such an idea of logic has been immanent to logical theory since its origins, i.e. at least since Aristoteles.

[4] The algebraic framework adopted here is close to that of [16].

Definition 2. Let $A = \langle\langle C_i\rangle_{i\in I}, \langle F_j\rangle_{j\in J}\rangle$ and $A' = \langle\langle C'_i\rangle_{i\in I}, \langle F'_j\rangle_{j\in J}\rangle$ be MSAs and let G be a function form the carrier of A to the carrier of A'. We shall say that G is a *homomorphism* if for any F_i there exists an F'_i such that

$$G(F_i(c_1, \ldots, c_n)) = F'_i(G(c_1), \ldots, G(c_n))$$

for any n-tuple $\langle c_1, \ldots, c_n\rangle$ from the domain of F_i.

Language can be viewed upon as MSA. In fact, language constitutes a very special kind of MSA; it is for example essential that it has a finite set of generators (words) and that any member of any sort is a "part" of a member of a certain distinguished sort (sentences). However, these properties are not relevant for our present investigations, and we shall therefore identify language with a general MSA.

Definition 3. A *language* is a MSA with a distinguished sort. The elements of the distinguished sort will be sometimes called *statements*. A sort of a language will be also called a *grammatical category* and its members *expressions* of the grammatical category. Operations will be called *grammatical rules*.

Now we define the concept of a polynomial function over a MSA. Polynomial functions are, roughly speaking, such functions which can be "built up" from operations. We shall see later what they are important for.

Definition 4. Let $A = \langle\langle C_i\rangle_{i\in I}, \langle F_j\rangle_{j\in J}\rangle$ be a MSA. Let us have for any $F_j (j \in J)$ a symbol $\mathbf{F}_j$. Let us, moreover, for any $i \in I$ have an infinite class of symbols $x_{i,1}, x_{i,2}, \ldots$ called *variables* of type i. For an n-tuple $\langle C_1, \ldots, C_n\rangle$ of sorts of A and a sort C_{n+1} we define the concept of a *polynomial symbol of the type* $\langle\langle C_1, \ldots, C_n\rangle, C_{n+1}\rangle$ *over* A as follows:

(i) if C_i is the jth constituent of the n-tuple S of sorts, then $x_{i,j}$ is a polynomial symbol of type $\langle S, C_i\rangle$ over A;

(ii) if $F_j (j \in J)$ is an operator with its domain equal $C_1 \times \ldots \times C_n$ and its range included in C_{n+1} and if $P_1, \ldots, P_n$ are polynomial symbols of the respective types $\langle S, C_1\rangle, \ldots, \langle S, C_n\rangle$, then the symbol $\mathbf{F}_j(P_1, \ldots, P_n)$ is a polynomial symbol of the type $\langle S, C_{n+1}\rangle$ over A.

If P is a polynomial symbol of a type $\langle\langle C_1, \ldots, C_n\rangle, C_{n+1}\rangle$, then we define a function F_P with its domain equal to $C_1 \times \ldots \times C_n$ and its range included in C_{n+1} in such a way that for any n-tuple $c_1, \ldots, c_n$ of elements of the respective sorts $C_1, \ldots, C_n$ the value $F_P(c_1, \ldots, c_n)$ is defined as follows:

(i) if P is $x_{i,j}$, then $F_P(c_1, \ldots, c_n) = c_j$; and

(ii) if P is $\mathbf{F}_j(P_1, \ldots, P_n)$,

$$\text{then} \quad F_P(c_1, \ldots, c_n) = F_j(F_{P_1}(c_1, \ldots, c_n), \ldots, F_{P_n}(c_1, \ldots, c_n)).$$

In general a function F is called *polynomial* if there exists a polynomial symbol P such that F is F_P.

It is obvious that any operation of A is polynomial.

Definition 5. Let $A = \langle\langle C_i\rangle_{i\in I}, \langle F_j\rangle_{j\in J}\rangle$ and $A' = \langle\langle C'_i\rangle_{i\in I}, \langle F'_k\rangle_{k\in K}\rangle$. We shall say that A' is a *polynomial extension of* A iff $J \subset K$ and for any $k \in K - J$ F_k is polynomial over A.

A polynomial extension thus results from an addition of some polynomial functions to the operations of the original algebra.

The view of language as of a MSA is a general one. It is not in contradiction with various more specific methodologies of formal description of natural language: all of them consider language as built from a lexicon by means of some rules of composition and all of them are thus in accordance with the present algebraic view.

However, not any algebra is plausible to deal with. We shall expose certain views which would favor MSAs of certain kinds and we shall discuss the question to what extent it is acceptable to consider NL as an algebra of such a kind. The kind of algebras we are going to expose are those which has become central to the symbolic description of NL as provided by formal logic.

2. THE MODEL-THEORETIC PERSPECTIVE

2.1 A general definition of the notion of a model

A powerful tool of the modern formal logic is the notion of model. The notion has been introduced by A. Tarski and since then it has become an integral and indispensable part of the apparatus of logic. Moreover, a model is based on an assignment of some "external" objects to expressions and thus fits well into our intuitive idea of semantics; meanings of expressions we also consider as extrenally tied to the expressions they are denoted by. This is the reason why model-theoretic results are often (although in majority of cases quite uncritically) employed in the explication of semantic notions.[5]

A model is an interpretation which satisfies a given theory. The notion of interpretation is usually defined for some very special class of formal languages, most often for the first-order logic. Within first-order predicate calculus an interpretation means an assignment of individuals of an universe to terms and of subclasses of (Cartesian powers of) the universe to predicates. The interpretation of logical connectives and quantifiers is considered as fixed once for all. This is a too special definition for our general view: we would like to define the concepts of interpretation and model quite generally, without any recourse to a special position of certain

[5] Montague (as well as other logically oriented semanticists) tacitly identifies semantic theory with a definition of a model-theoretic interpretation of NL. It seems that such a step needs further substantiation, as it is in no way clear why the explication of the notion of meaning should consists in a definition of a Tarskian model; however, we have not the opportunity to investigate into this question here. For a detailed analysis of this problem see [25].

elements of language (as e.g. the logical operators).[6] What seems to be essential for an interpretation is that it is always defined in a "compositional" way[7]: its value for a complex expression is a function of interpretations of parts of the complex. Let us articulate this formally.

Definition 6. Let $A = \langle\langle C_i\rangle_{i\in I}, \langle F_j\rangle_{j\in J}\rangle$ be a MSA and let G be a function with its domain equal to the carrier of A. We shall say that G is *compositional* iff for any F_j there exists a function F'_j such that

$$G(F_j(c_1, \ldots, c_n)) = F'_j(G(c_1, \ldots, G(c_n))$$

for any n-tuple $\langle c_1, \ldots, c_n\rangle$ from the domain of F_j.

Theorem 7. Let $A = \langle\langle C_i\rangle_{i\in I}, \langle F_j\rangle_{j\in J}\rangle$ be a MSA and G a compositional function, then $A' = \langle\langle C'_i\rangle_{i\in I}, \langle F'_j\rangle_{j\in J}\rangle$, where C'_i is $G'C_i$ (i.e. the image of C_i under G), and F'_j is the function from the previous definition, is a MSA and G is a homomorphism of A into A'.

Proof. Evident. □

We can see that an interpretation can in general be considered as a homomorphic mapping of the language into a MSA.

Definition 8. Let L be a language. A homomorphism of the MSA L into a MSA S will be called an *interpretation of* L is S.

In fact, what is important for an interpretation is that it interprets certain sentences in a special way, i.e. by some distinguished entity. An interpretation is then a *model* of a class of statements (of a theory) if it interprets all and only members of the class by the distinguished element.

We have no possibility to develop a theory of many-sorted algebras in detail here. We shall restrict ourselves to the investigation of the question when a given language has an applicative interpretation. A more complicated connected problem is when a given theory has an applicative model, and when any theory in a language which has a model has an applicative model. These questions are, of course, more problematical, however, they can be inquired into by the same means as those exposed here.

2.2 Applicative interpretations

There are usually many kinds of interpretations of a given language. It appears to be plausible to pick up the "most convenient" of them, usually such one the

[6] Also more general model theories, such as Kemeny's ([17]) model theory for Church's typed lambda-calculus, are not general enough for us. The problem is that even Church's system constitutes a quite special kind of MSA. This we shall see in clarity later.

[7] The so-called principle of compositionality is usually attributed to Frege; however cf. [16]. In any case the principle has become a constitutive postulate of the whole modern formal semantic and formal logic.

operations of which are in certain sense simple. The elements of any algebra are sets; and it seems to be plausible for the operations to be set-theoretical functional applications. Let us distinguish the class of MSA which fulfil this condition.[8]

Definition 9. A MSA A is called *applicative* iff there exists an injective function $FUNCT$ with its domain equal to the set of the operations of A, its range included in the set of sorts of A, and such that if F is an operation of A with its domain equal to $C_1 \times \ldots \times C_n$ and its range included in C_{n+1}, then $FUNCT(F) = C_i$ for some i from 1 to n, and, moreover, C_i is the class of functions from $C_1, \ldots, C_{i-1}, C_{i+1}, \ldots$ $\ldots, C_n$ into C_{n+1} such that for any elements $c_1, \ldots, c_n$ of $C_1, \ldots, C_n$, respectively, $F(c_1, \ldots, c_n)$ equals the functional application of c_i to $c_1, \ldots, c_{i-1}, c_{i+1}, \ldots, c_n$. The category C_i is then called the *functor category* of F; the categories $C_1, \ldots, C_{i-1}$, $C_{i+1}, \ldots, C_n$ are called *argument categories* of F. $C_1, \ldots, C_n$ are called the categories *dealt with by F*.

Definition 10. A language is called *applicative* if it has an applicative interpretation.

If I is an applicative interpretation of a language $\boldsymbol{L}$ in a MSA A, then under the *I-functor category* (an *I-argument category*) *of a rule* R we shall understand the category mapped by I on the functor (an argument) category of the operation of A corresponding to R.

Now the question arises, if any interesting class of languages can be considered as applicative, and especially, if NL can. To be able to answer this question we first have to find some simple criterion of applicativity.

Let us assume that we have an applicative MSA. Then any rule is assigned a functor category. Can such an assignment of categories to rules as the functor categories be wholly arbitrary? Surely not: for example no category can be an functor category of a rule of which it is also an argument category. This follows from the fact that a function cannot be a member of its own domain. Moreover, a category C cannot be an argument category in a rule in which C' is a functor category if C' is an argument category of a rule of which C is the functor category. This indicates that in language which allows for applicative models there has to exist a special kind of assignment of categories to rules.

Theorem 11. If a MSA is applicative, then we can index its categories in such a way that any rule combines expression of a category with an index $A/B_1, \ldots, B_n$ with expressions of the categories $B_1, \ldots, B_n$, respectively.

Proof. As any function must neither be a member of its own domain, nor a component of such a member, the categories of L can be partially ordered in such a way

[8] For the usual definitions of the notion of model within various propositional and predicate calculi applicativity is implicit. Therefore it seems to be rather our applicative model what corresponds to the traditional notion of model in the closest way.

that for any rule its functor category is greater that all its argument categories. However, this is clearly equivalent to the possibility of the indexing in question. □

As there is a one-to-one correspondence between the functor categories and rules of a MSA and the categories and rules of any its homomorphic image we have

Corollary 12. If a language is applicative, then we can index its categories in the way described in Theorem 11.

It can be shown, that also the converse holds.

Theorem 13. If for a language there exists the indexing described in Theorem 11, then it is applicative.

Proof. Let us call those categories which are assigned noncompound indices *ground*. Let us define the function I in such a way that to any member of any ground category it assigns the member itself (or whatever), and to an expression E of a category $A/B_1, \ldots, B_n$ it assigns such a function from $I'B_1 \times \ldots \times I'B_n$ into $I'A$[9] that for the relevant grammatical rule R

$$I(R(E_1, \ldots, E_{i-1}, E, E_{i+1}, \ldots, E_n)) =$$
$$= I(E)\,(I(E_1), \ldots, I(E_{i-1}), I(E_{i+1}), \ldots, I(E_n)) .$$

Then I is an applicative interpretation. □

The existence of such an indexing, which we shall for historical reasons call *categorial indexing*, is thus a necessary and sufficient condition of applicativity.[10]

Let us now try to investigate the common languages of formal logic as for their applicativity. First, let us consider the language of the ordinary propositional logic. There are three categories ($PROP$, propositions, O_1, one-place operators, and O_2 two-place operators) and two grammatical rules (R_1: $O_1 \times PROP \to PROP$ and R_2: $O_2 \times PROP \times PROP \to PROP$). We can see that for any rule there is a category such that it is dealt with only by the rule: O_1 is dealt with only by R_1, O_2 by R_2; O_1 thus can be assigned the index $PROP/PROP$ and O_2 $PROP/PROP, PROP$. This means that whatever is the interpretation I of the elements of $PROP$, expressions of O_1 can be interpreted as functions from $I'PROP$ into $I'PROP$ and those of O_2 as functions

9 If F is a function and X a subset of its domain, then $F'X$ denotes the set of all $F(x)$ such that x is an element of X.

10 The view of grammar as consisting of the applicative rules and connected with the categorial indexing has been introduced by Ajdukiewicz ([1]) and then elaborated in a systematic way by Bar-Hillel ([2]) and Lambek ([18]). Bar-Hillel writes: "Each sentence which is not an element is regarded as the outcome of the operation of one sub-sequence upon the remainder. That sub-sequence which is regarded as operating upon the others will be called an operator the other its arguments". (p. 65) This exposes another motivation of the approach: the rules are considered as certain ways of "saturation" of "unsaturated" expressions (cf. [15]). The methodology of description of the grammar of NL by this approach is often called a *categorial grammar*. For a detailed exposition of its roots and history see [8].

from $I'PROP \times I'PROP$ into $I'PROP$. If, as usual, $I'PROP$ is the set $\{T, F\}$, the interpretations of one- or two-place operators may be the usual truth functions.

If we add the gremmatical categories IND of subject terms and $PRED_n$ (for any natural n) of n-ary predicates, and the rules $R_{3,n}$: $PRED_n \times IND \times \ldots \times IND \rightarrow$ $\rightarrow PROP$ to the propositional calculus, we gain the first-order predicate calculus without quantifiers. It is easy to see that also this language is applicative: $PRED_n$ can be assigned the index $PROP/IND, \ldots, IND$ (n times). We then interpret n-ary predicates as functions from n-tuples of elements of $I'IND$ into $I'PROP$, i.e. in the most usual case as n-ary relations among individuals.[11]

Let us now try to add quantifiers to the language of first-order predicate calculus without quantification; let us for the sake of simplicity restrict ourselves to the monadic case, i.e. to the language with only unary predicates. We thus add the category $QUANT$, the grammatical rule R_4: $QUANT \times PRED_1 \rightarrow PROP$ and index $QUANT$ by $(PROP/(PROP/IND))$. However, this will not do. The reason is that in the resulting language we can combine quantifiers only with predicate *constants*, as there is no rule which would produce a complex predicate expression. And we would surely want to use quantifiers with also, for example, *conjunction* of predicate constants. To accomplish this we would have to add rules R_5: $O_1 \times PRED_1 \rightarrow$ $\rightarrow PRED_1$ and R_6: $O_2 \times PRED_1 \times PRED_1 \rightarrow PRED_1$; and by such an addition there is no more way to perform the categorial indexing and thus to consider the language as applicative. We can thus see that the requirement of applicativity is too strong to be fulfilled even by such a basic logical system as the first-order predicate calculus.

2.3 Combinatorial languages

The identification of operations of language with functional application is thus untenable. However, the interpretation of expressions by functions need not be done away in the same time.

Let us realize that application is not the only operation which can be done with functions. We can, for example, *compose* functions. Would it not be possible to weaken the demands on the rules in the sense that they can be not only applications, but also compositions or whatever, while members of the semantic algebra remain functions?

It appears that in the case of the first-order predicate calculus this *is* possible. We can identify the rule R_5 with the composition of the function assigned to the one-place sentential operator with the function assigned to the unary predicate (and similarly for R_6). It is necessary for this to be possible that for any $p \in PRED_1$, $o \in O_1$ and $s \in IND$ it holds that $R_1(o, R_{3,1}(p, s)) = R_4 R_5((o, p), s)$; but this is the way matters really are in first-order predicate calculus.

[11] We identify a set with its characteristic function in the usual way.

This indicates that we can add some rules to an applicative language while retaining its applicative interpretation. However, they have to be "reasonable" functions, and *reasonable* in this context means nothing other than *polynomial*.

Definition 14. A language is *combinatorial* if it is a polynomial extension of an applicative language.

Theorem 15. Let $\boldsymbol{A}$ be an applicative language and I its interpretation in an applicative MSA $\boldsymbol{B}$. Let $\boldsymbol{A}'$ be a polynomial extension of $\boldsymbol{A}$. Then there exists a polynomial extension $\boldsymbol{B}'$ of $\boldsymbol{B}$ such that I is an interpretation of $\boldsymbol{A}'$ in $\boldsymbol{B}'$.

Proof. The statement is a consequence of Theorem 7.3 of [16]. □

This means that we can add polynomial rules without affecting interpretations. This is to say that a function is an interpretation of a combinatorial language if it is an interpretation of its "applicative part". A language is thus combinatorial iff there exists a categorial indexing of its categories such that any rule of the language is polynomial over the applicative part.[12]

There is also another interesting characterization of the combinatorial languages. We have characterized applicative systems by that their categories can be indexed in such a way that any rule combines expression of a category with the index $(A/B_1, \ldots, B_n)$ with expressions of the categories $B_1, \ldots, B_n$. Let us write $(B_1 \rightarrow \ldots \ldots (B_n \rightarrow A) \ldots)$ instead of $(A/B_1, \ldots, B_n)$; we thus associate a logical formula with any category. Then there is a rule combining expressions of the categories $C_1, \ldots, C_n$ into an expression of the category C_{n+1} if the "logical index" of C_{n+1} "follows from the logical indices of $C_1, \ldots, C_n$ by *modus ponens*". Regarding applicative systems this is only a cumbersome articulation of what has been articulated quite simply before; however, this view helps us to characterize the combinatorial languages in a surprising way. The point is that in combinatorial languages grammatical rules correspond to derivational rules of certain implicational calculus: there is a rule combining expressions of the categories $C_1, \ldots, C_n$ into an expression of the category C_{n+1} if the "logical index" of C_{n+1} follows from the "logical indices" of $C_1, \ldots, C_n$. Thus, for example, a rule may combine expressions of a category A/B with expressions of a category B/C into exprcssions of the category A/C, as it holds that $(B \rightarrow A)$, $(C \rightarrow B) \Rightarrow (C - A)$.[13]

[12] What is closely connected with the notion of polynomiality is Church's apparatus of lambda-conversion. In fact any polynomial can be viewed upon as an open lambda-term; a language thus can be considered as combinatorial iff any its rule can be assigned an (open) lambda-term. Church's formulation of the simple theory of types can thus in fact be considered as a "complete" combinatorial language. Another way of looking at polynomials is through Curry's and Feys' theory of combinators (see [12]), which is proven to be equivalent to the lambda-calculus. The "combinatorial turn" in categorial grammar has been taking place recently — see e.g. [28].

[13] For a detailed exposition of this problem see [4]. Cf. also [29].

3. THE FORMALIZATION OF NATURAL LANGUAGE

3.1 Extensionality vs. intensionality

The logistic systems explicating the semantics of NL such as the so-called *Montague Grammar*, based on the investigations of Montague ([20]), or Tichý's *Transparent Intensional Logic* (see e.g. [30]), are based on Church's typed lambda-caculus ([10]) and thus in fact on a combinatorial language. In this logical calculus the categories are defined as follows: there are two ground categories, o (propositions) and ι (individuals): and if α and β are categories, then $(\alpha\beta)$ is also a category (in contrast to the above used Ajdukiewicz's notation, Church uses $(\alpha\beta)$ instead of α/β). Complex expressions are constituted by means of application and lambda-abstraction; we know that this is equivalent to having combinators as grammatical rules. (As any closed lambda-term is expressible as an application of suitable combinators to constants of the lambda-term, we can also see that variables are not necessary; they only make everything technically simpler).

Church has introduced what is usually called *postulates of extensionality*; they in fact guarantee that (i) any two statements with the same truth value are intersubstitutive *salva veritate* for any theory admissible in his system; and (ii) that any two predicates which for any arguments yield the same value are intersubstitutive *salva veritate* for any such theory. However, as it has soon become clear, the interpretation intended by Church, i.e. the interpretation of statements by truth values and predicates by classes of (n-tuples of) actual individuals is inadequate with respect to NL. Such predicates as "human" and "featherless biped"[14] are to be of the category $(o\iota)$ in Church's system, and as the classes of actual individuals to which they truthfully apply are identical, they coincide. However, there is surely a true statement which becomes false if we substitute *human* for *featherless biped* in it (for example the statement *A shaved circus bear might be a feartherless biped*), and thus Church's extensionality postulates fail to render the factual character of NL.

This fact has been clear at least since Carnap ([6]) and it has been under a thorough discussion since sixties. The seemingly simpler way out of this mess, to drop the postulate of extensionality, leads to the loss the plausible set-theoretical interpretation of the language. Therefore another way has been largely exploited: to schematicize NL by means of an extensional formal language, however, to accopmish the schematization in such a way that statements are not considered as of type o, i.e. they are not interpreted by truth-values, but they are rather assigned a compound type interpreted by means of functions from objects of a newly introduced type, the type of *possible worlds*, into the set of truth-values.

The best example of the strategy is constituted by Tichý's *Transparent Intensional Logic* (TIL). Tichý modifies Church's logical system in that he adds new simple categories ω (possible worlds) and τ (time moments), and assigns the category $((o\tau)\ \omega)$

[14] This is a famous example from [6].

to NL sentences. Unary predicates then turn out to be of the category $(((o\iota)\,\tau)\,\omega)$ etc. Tichý thus works with a calculus which is in fact extensional, he, however, relates it to NL in such a way that NL comes out as non-extensional.[15]

3.2 Predication and nominalization

There is also another problem which the "categorial" approach to NL has to face. There is a strong intuition indicating that two expressions of different categories have to be interpreted by one and the same item. If this were true, the view of interpretation as a homomorphism would clearly become untenable.

Let us consider an adjective and its nominalization, say *clever* and *cleverness*. To say that one is clever means to say that cleverness belongs to him. From this it seems to follow (see the detailed considerations of [24]) that the two words have one and the some denotation. This is also the way in which matters are in Church's formulation of the theory of types. *Clever* denotes a function from individuals to truth values and so does *cleverness*; predicates applicable to *cleverness* then denote functions from these functions to truth values.

However, there are predicates which are evidently predicable both of individuals proper and of nominalized predicates. This is the case of, say, *good*. Schematically, if we denote the class of individual terms proper as I, the class of individual predicates by P, the class of predicates predicable of nominalized individual predicates as P' and the class of statements as S, predication can be viewed upon as two rules: R_1: $P \times I \to S$ and R_2: $P' \times P \to S$. However, as soon as we accept that predicates of P' apply to individuals as well, we have also the rule R_3: $P' \times I \to S$. It is not difficult to see that any grammar containing R_1, R_2 and R_3 can be neither applicative (in the sense exposed above), nor combinatorial. This phenomenon has been studied by Cocchiarella ([11]), he adds an apparatus of nominalization to predicate logic. It is clear that as soon as we can turn any first-order predicate into an individual, we can turn any second-order predicate into a first-order, and hence into an individual and so on; thus having introduced nominalization the whole theory of types collapses into the first-order case. In this way, however, the whole idea of type-theoretical treatment of NL seems to be questioned. Chierchia's application of Cocchiarella's ideas to concrete linguistic problems (see [7]) as well as Turner's ([32]) elaboration of another similar system with nominalization (based on Scott's models for type-free lambda-calculus – see [26]) indicate that also systems without the strict typization are worth investigation.

[15] The matters seem to be more complicated with respect to Montague; his logical system seems to be really overtly nonextensional. However, the semantic interpretation of Montague Gramar is based on a tacit "type-lifting" which results in a model theory of the same form as that of Tichý. It is characteristic that Janssen in his extensive monograph on Montague Grammar ([16]) introducing the Montague's logical system first introduces a language Ty 2, very close to TIL, and then he defines Montague's logic in its frames.

3.3 Indirect description by means of first-order logic

We have seen that even the complete combinatorial grammar, i.e. the whole theory of types is disputable as for its adequacy for the purpose of schematicizing of NL. On the other side, we would like to have a schematization of NL in frames of a much simpler logical system. The reason is that we have no effective proof apparatus for predicate logic of an order higher than the first one. For the first-order logic, we, on the other side, do have a method which has allowed us to implement deduction on a computer.

So far we have considered logic as a more or less direct schematization of NL. This means that we have considered any constant of a logical calculus to symbolize a NL expression and any grammatical rule of such a calculus to represent a corresponding rule of NL. This is the way in which logic can be made sense of; however, we may accept that a logical formula need not schematicize directly the expression we are considering, but some its paraphrastic forms. This is to say, considering a statement of the form

John loves Mary faithfully

we may consider it to be schematicized by the first-order formula

$$\exists \boldsymbol{E}(Event(\boldsymbol{E})\,\&\,Loving(\boldsymbol{E})\,\&\,Ac(John,\,\boldsymbol{E})\,\&\,Obj(\mathrm{Mary},\,\boldsymbol{E})\,\&\,Faithful(\boldsymbol{E}))$$

which in fact schematicizes the paraphrase

There is an event of loving such that John is its actor ,
Mary its objective, and it is faithful.

This example indicates in what way an adverb, which is usually conceived as a predicate of a higher order, can be turned into a first-order predicate by means of this "paraphrastic strategy".

The introduction of events, illustrated by this example, is the idea of Davidson ([13]). Recently it has been picked up and further elaborated in [21]. It shows that while on one side even very complex logical systems seem insufficient for the purposes of schematization of NL, on the other side we can accomplish interesting results with quite simple logical means if we use them adjointly.[16]

4. TOWARDS A WHOLLY ADEQUATE FORMALIZATION OF THE SEMANTIC ASPECT OF NATURAL LANGUAGE

An adequate formalization of the semantic aspect of NL is still something to be inquired into. We have seen in the previous chapter what problems could be connected with the application of the common "ready-made" logical systems (which have

[16] See also [23].

arisen from considerations of various restricted parts of NL) to the NL in its entirety. It is surely useful to try to find ways how to handle as much of NL as possible in their frames, as the achievements of logical theory are valuable; however, on the other side, seeking for the real character of the semantic aspect of NL we cannot use a present form of logic as a mould into which NL has to be squeezed cost what it may. As we have stated elsewhere ([24]), theory is not to "repair", but rather to *explicate*.

The import of logic for semantics consists besides other in the fact that logic has indicated in what way a quite simple semantical account of the core of NL can be given. Sentences of the ordinary subject/predicate structure. their conjuncts by means of ordinary connectives, even the basic modal and temporal adverbs – all of this can be schematicized by existing languages of formal logic and in this way furnished by suitable model-theoretic interpretation. There are, however, parts of NL which cause serious problems to any existing logical system.[17] To be able to give a proper logical account of all parts of NL including the peripheral ones, we have to perform diligent empirical considerations of the language, i.e. a full-fledged linguistic analysis.

Results of theoretical linguistics relevant for the formalization of semantics, which have been achieved by all the various linguistic schools up to now, are, however, often presented in the form which seems to be indigestible for a logician. A logical interpretation of an expression is usually a function of the kind which has been described above; linguistics, on the other side, often presents the "semantic content" of an expression in a form of a diagram, typically a tree, connecting linguistic items, present in thee expressions, by labelled arcs. Is there a bridge between these two views of meaning?

The extensional account of semantics, as well as the intensional one in its pure form, is based on the assumption that the internal syntactic structure of an expression is irrelevant from the semantic point of view. This assumption has, however, been challenged seriously in the recent years: many new formal theories of meaning which have appeared in the present decade accept a tacit assumption that meaning is structured in a way corresponding to the way of structuring of the expression by which it is denoted. Roots of such an assumption can be found in Carnap's notion of intensional isomorphism (see [6]); as a prominent example of this kind of theory we can name the situation semantics ([3]) in which meanings are modeled by means of situations, which are set-theoretical objects, decomposable into a relation (corresponding to a verb), objects to which the relation apply (corresponding to the participants of the verb), location (corresponding to a local adverb) etc. In [31] Tichý presents a comprehensible account of what the proper place of a notion of a *construction*, a language independent entity expressed by an expression and mirroring its structure, in frames of the Fregean logic could be.

[17] Such problems are numerous and their extensive definition keeps taking place in semantic literature. Let us mention at least the problem of the verbs of intensional attitudes ([5], [19], [22], and also [27]).

However, in connection with this new approach to semantics it is necessary to keep in mind the fact that the correspondence of the structure of an expression and that of the meaning denoted by it cannot be absolute. If it were, semantic analysis would be a mere triviality. Besides this, and this is the main reason, it is clear that expressions which differ in syntactic structure can be synonymous. This fact can be accounted for by that it is not the surface structure what is relevant for the structure of meaning, but rather a "deep" structure. From the semantic point of view it is thus not the grammatics what is relevant, but rather a "tectogrammatics" (for the notion of tectogrammatics see [27]).

This is the point at which the logical theory of interpretation and the linguistic theory of content seems to meet each other. There is, however, much to be done, in order to achieve a really firm connection between them. In order to build a formal theory of semantics which would be not only technically plausible, but also empirically adequate, this seems to be nevertheless essential.

ACKNOWLEDGMENT

The author thangs to P. Sgall and P. Materna for valuable critical comments.

(Received October 13, 1988.)

REFERENCES

[1] K. Ajdukiewicz: Die syntaktische Konexität. Studia Philosophica *1* (1935), 1—27.

[2] Y. Bar-Hillel: A quasi-arithmetical notation for syntactic description. Language *XXIX* (1953), 47—58.

[3] J. Barwise and J. Perry: Situations and Attitudes. MIT Press, Cambridge 1983.

[4] J. van Benthem: Categorial grammar and type theory. Linguist. Philos. (to appear).

[5] J. C. Bigelow: Believing in semantics. Linguist. Philos. *2* (1978), 101—144.

[6] R. Carnap: Meaning and Necessity. Univ. of Chicago Press, Chicago 1957.

[7] G. Chierchia: Topics in the Syntax and Semantics of Infinitives and Gerunds. Ph. D. Dissertation, Univ. of Massachusets, Amherst 1984.

[8] C. Casadio: Semantic categories and the development of categorial grammars. In: Categorial Grammars and Natural Language Structures (T. Oeherle et al., ed.). Reidel, Dordrecht 1988, 95—123.

[9] N. Chomsky: Syntactic Structures. Mouton, The Hague 1957.

[10] A. Church: A formulation of the simple theory of types. J. Symbolic Logic 5 (1940), 56—68.

[11] N. B. Cocchiarella: Conceptualism, realism and intensional Logic. Manuscript.

[12] H. B. Curry and R. Feys: Combinatory Logic. North Holland, Amsterdam 1957.

[13] D. Davidson: Essay on Actions and Events. Claredon Press, Oxford 1980.

[14] G. Frege: Begriffsschrift. Verlag von H. Nebert, Halle, Saale 1979.

[15] G. Frege: Funktion und Begriff. Ein Vortrag gehalten in der Sitzung vom 9. 1. 1981 der Jenaischen Gesellschaft für Medizin und Naturwissenschaft. Reprinted in Logik Texte (K. Berka and L. Kreisel, eds.), Akademie-Verlag 1971.

[16] T. M. V. Janssen: Foundations and Applications of Montague Grammar. Mathematisch Centrum, Amsterdam 1983.

[17] J. G. Kemeny: Models of Logical Systems. J. Symbolic Logic *13* (1948).
[18] J. Lambek: The mathematics of sentence structure. Amer. Math. Monthly *LXV* (1958), 154—170.
[19] P. Materna: On understanding and believing. Prague Studies in Math. Lingustics *8* (1983), 211—218.
[20] R. Montague: Formal Philosophy: Selected Papers of R. M. (R. Thomasson, ed.), Yale Univ. Press, Yale 1974.
[21] T. Parsons: Underlaying events in the logical analysis of English. In: Actions and Events (E. LePore and B. P. McLaughlin, eds.), Blackwell 1985.
[22] J. Peregrin: A contribution to the theory of propositional attitudes. Prague Bulletin of Mathematical Linguistics *48* (1987), 13—36.
[23] J. Peregrin: Přirozený jazyk, PROLOG a „události" (Natural Language, PROLOG and "events"). Proc. of the AI' 88 Symposium, Prague 1988 (in Czech).
[24] J. Peregrin: Theory of types: good servant bad master. Prague Studies in Math. Lingustics (to appear).
[25] J. Peregrin: A formal theory of meaning. In preparation.
[26] D. Scott: Continuous lattices. In: Toposes, Algebraic Geometry and Logic (F. W. Lawrence, ed.), Springer-Verlag, Berlin 1972.
[27] P. Sgall, E. Hajičová and J. Panevová: The Meaning of the Sentence in its Semantic and Pragmatic Aspects. Academia, Prague 1986.
[28] M. J. Steedman: Combinators and grammars. In: Categorial Grammars and Natural Language Structures (T. Oeherle et al., eds.), Reidel, Dordrecht 1985, 417—442.
[29] A. Strigin: Logic for syntax. In: Proc. '87 Debrecen Symposium on Logic and Language (I. Ruzsa, A. Szabolcsi, eds.), Budapest 1987, pp. 163—170.
[30] P. Tichý: Logic of temporal discourse. Linguist. Philos. 3 (1980), 313—369.
[31] P. Tichý: Constructions. Philos. Sci. *58* (1986), 514—543.
[32] R. Turner: Montague semantics, nominalization and Scott's domains. Linguist. Philos. *6* (1983), 259—288.

RNDr. Jaroslav Peregrin, ORGATEX, Veverkova 1631, 500 05 Hradec Králové. Czechoslovakia.