ELIMINATION OF QUANTIFIERS
IN THE SEMANTICS OF NATURAL LANGUAGE
BY USE OF EXTENDED RELATION ALGEBRAS (1)

by PATRICK SUPPES

1. INTRODUCTION

A plethora of proposals besieges seekers after an adequate semantics of natural language. The special twist of my proposal is to offer simplicity, not generality. We need, so it seems to me, systematic semantic analyses of restricted but substantive fragments of natural language. Such analyses should have the same broad aims as restricted models in other areas of knowledge: simplicity of formulation, ease of comprehension, and computational efficiency.

The present work is an outgrowth of my earlier work on the semantics of context-free fragments of natural language (Suppes, 1973a). The arena is narrower, but my intention is to enter more deeply into the details. One egocentric way of putting my objective is that I propose an analysis in terms of what I call extended relational algebras of the semantics of a large part of the or-

(1) The ideas developed here were first outlined in my Stanford course of lectures on mathematical linguistics in the winter term of 1973. They were almost crystallized into their present form in a joint seminar I gave with Paul Grice in Berkeley during the spring term, 1975. He would undoubtedly not wholly agree with everything that I say here, although in response to his sharp questions, and those of George Myro and Richard Warner as well, I made a valiant effort at persuasion. I have benefited recently from a lively exchange with Robert Smith and Freeman Rawson. Their trenchant remarks on the next-to-final draft saved me from several blunders and obscurities. This research has been supported in part by U.S. National Science Foundation Grant NSF-EC-43997.
dinary-language examples of quantifiers in my logic text (Suppes, 1957). Thus, I want to analyze not only All men are animals, but also Some sophomores date juniors, Every man loves some woman, Some men look at every woman, or Every number is less than some number. It is, of course, easy to construct examples of mathematical sentences that fall outside the intended range of analysis, e.g., *For all distinct points x, y, and u, if x, y, and u are not collinear, then there is a point v such that the line determined by x and y is parallel to the line determined by u and v* (2). But the extent of ordinary usage covered by the semantic calculus of extended relation algebras is substantial, mainly because complex sequences of quantifiers are rare in natural language.

I hasten to add there are many other important features of natural language that the present theory gives no account of—perhaps the most salient example would be the semantics of adverbs. On the other hand, extensions of what I develop here could handle parts of the semantics of tense in verb systems, and also some essential semantic features of anaphora.

The advantages of not needing a full Zermelo-like hierarchy of sets for the model structures of ordinary sentences are obvious. In a general way the algebraic push for efficiency of computation by avoiding a hierarchy is probably a rather widely acceptable idea.

Other aspects of the “algebraic” semantics proposed here are much more controversial. Linguists especially will not like my insistence that such expressions as every man, all freshmen, or some dogs are not noun phrases. Furthermore, these phrases as such do not, in my approach, have a denotation, contrary, for example, to the semantics proposed by Grice or Montague. Indeed, the quantifier words every, all, some, etc., do not denote at all in my approach but function as control structure words—to use a terminology favored by computer scientists (4). To be more ex-

---

(2) From a formal standpoint, a better example, taken from Tarski (1941), that cannot be expressed in the calculus of relations is this: *For every x, y, and z there is a u such that xRu & yRu & zRu.*

(3) For a sophisticated discussion of control structures in natural language, see Smith and Rawson (in press).
explicit, they control the selection of semantic functions in a way that will be explained in detail later.

My central point is that the fixation on first-order logic as the proper vehicle for analyzing the "logical form" or the "meaning" of sentences in a natural language is mistaken. The semantics of first-order logic is ideal for the expressions of that logic. The syntax and semantics fit each other as subtly as could be desired, *pace* Gödel’s completeness theorem. The same snug fit of syntax and semantics should be our objective for natural language, but such a fit will not be obtained by rigidly holding to the apparatus of first-order logic to frame the semantics of natural languages. Individual variables do not occur in most natural utterances. A semantical approach that requires their appearance is bound to fit the syntax in an awkward way. Much of what I am saying has been recognized de facto at least in the special case of the standard A, E, I, and O sentences that occur as premises or conclusions of syllogisms. What I attempt to show here is that the algebraic approach can be extended considerably beyond these limited forms to a fairly rich body of natural utterances.

On the matter of the denotation or lack of it of *every man*, and the corresponding linguistic production rules for sentences and noun phrases, we can prove in a simple way that the conventional production rules will not even permit a semantic account of the syllogism at the level of Boolean algebra, i.e., at the level of subsets of the domain, but require what I consider an artificial escalation of type (see Section 4).

Exhaustive discussion of the variant positions that have been taken toward the denotation or reference of *every man* has been given by Geach (1968), but he passes over, without much analysis, a position close to that advocated here (p. 13). Since the introduction of quantifiers as explicit logical notation by Frege, the separation of semantics from the syntax of natural language seems to have widened. I see no reason for this. The more the logical apparatus of semantical analysis is separated from the actual syntactical forms of natural language, the less reason there is to accept its correctness as an analysis of what has been said.
Although Frege did not develop a closely linked syntax and semantics of natural language — and was indeed suspicious of the logical coherence of natural language — his informal view of the logical role of quantifier words in natural language is close to what I am advocating. Here is a significant passage (Frege, 1960, p. 48) referred to by Geach:

'It must here be remarked that the words 'all', 'any', 'no', 'some', are prefixed to concept-words. In universal and particular affirmative and negative sentences, we are expressing relations between concepts; we use these words to indicate the special kind of relation. They are thus, logically speaking, not to be more closely associated with the concept-words that follow them, but are to be related to the sentence as a whole. It is easy to see this in the case of negation. If in the sentence 'all mammals are land-dwellers' the phrase 'all mammals' expressed the logical subject of the predicate are land-dwellers, then in order to negate the whole sentence we should have to negate the predicate: 'are not land-dwellers'. Instead, we must put the 'not', in front of 'all'; from which it follows that 'all' logically belongs with the predicate.

As Frege hints at, some change in a tradition that goes back to the Alexandrian grammarians is required in the linguistic classification of quantifier words. This reclassification is already presaged in the treatment of quantifiers in first-order logic where they are not categorized as terms or parts of terms. By modifying both the logical and the linguistic traditions, an approach can be given, so I claim, that brings completely together the syntax and semantics of sentences containing quantifier words — not all such sentences, but many of the garden-variety ones.

I shall proceed in the following fashion. Section 2 is devoted to a quick overview of my conception of semantic trees for context-free languages. Section 3 develops the restricted semantics of extended relation algebras. Section 4 is concerned with the proof that the kind of grammatical production rules often used by linguists for quantifiers, in contrast to the ones given here, lead to an undesirable escalation of logical type in the underlying set-theoretical semantics.
2. Semantic Trees

The ideas central to model-theoretic semantics in mathematical logic go back to Frege. They have been developed extensively by the work of Tarski and his students, but most of model theory is not concerned with natural language. The semantical approaches developed by linguists or others whose viewpoint is that of generative grammar have been lacking the formal precision of model-theoretic semantics. My objective here and in other writings has been to combine the viewpoint of model-theoretic semantics, on the one hand, and generative grammar, especially the work of Chomsky, on the other.

A number of examples will give an intuitive idea of the intended application of model theory. Let us take as an instance the simple noun phrase *square table*. That noun phrase can be represented by a simple tree (Figure 1), where NP = noun phrase, N = noun, and Adj = adjective.

![Figure 1](image)

This grammar has (except for insertion of lexical items) as yet only one production rule: NP → Adj + N. How is the semantics of this noun phrase to be formalized? With each production rule of the grammar we associate a semantic function, and thus we may convert each derivation tree to a *semantic tree* by assigning a denotation to each node of the tree. The denotation of each node of the tree is shown in Figure 2 after the colon following the label of the node.

![Figure 2](image)
The semantic tree is obtained by the assignment of (i) denotations to terminal words or phrases, in this case $S$ for the set of square things and $T$ for the set of tables, and (ii) semantic functions to production rules, in this case the identity function for the lexical rules and intersection for $NP \rightarrow Adj + N$. The word order in the corresponding French example *table carrée* (Figure 3) is reversed. The denotation of the root of the tree

![Figure 3](image)

is just the same. The semantic parts of the French and English trees are more alike than the grammatical parts.

Let us next consider the analysis of a simple sentence. The semantic tree of *John loves Mary* has the following form (Figure 4):

![Figure 4](image)

Note that in this tree $L$ is the binary relation of loving. Perhaps the most interesting denotation is that of the VP node. The denotation of this node is the set consisting of the image of Mary under the converse of loving --- $L^{-1}[m]$ --- or, in more intuitive language, the set of persons that love Mary. The denotation of the root of the tree I take to be what I term the Frege function defined as follows:

$$[F \varphi (A,B)] = \begin{cases} T & \text{if } \varphi (A,B) \\ F & \text{otherwise.} \end{cases}$$
I give a brief overview of the systematic ideas back of these examples. First, a structure \( G = \langle V, N, P, S \rangle \) is a phrase-structure grammar if and only if \( V \) and \( P \) are finite, nonempty sets, \( N \) is a subset of \( V \), \( S \) is in \( N \), and \( P \subseteq N^* \times V^* \) where \( N^* \) is the set of all finite sequences whose terms are elements of \( N \), and \( V^* \) is \( V^* \) minus the empty sequence. The grammar \( G \) is context-free if and only if \( P \subseteq N \times V^* \). In the usual terminology, \( V \) is the vocabulary, \( N \) is the nonterminal vocabulary, \( S \) is the start symbol of derivations or the label of the root of derivation trees of the grammar, and \( P \) is the set of production rules.

This formal definition of a grammar may be illustrated by an example, call it \( G_1 \), that will be useful in the subsequent discussion. Note that \( G_1 \) generates, among other things, the classical A, I, E, and O propositions. The set \( P_1 \) of production rules, minus the lexical rules for introducing terminal vocabulary, is:

\[
\begin{align*}
S & \rightarrow UQ + NP + VP / EQ + NP + VP / NQ + NP + VP \\
NP & \rightarrow N \\
VP & \rightarrow TV + UQ + NP / TV + EQ + NP / TV + NQ + NP / IV \\
& \quad \text{Cop} + NP / \text{Cop} + \text{Adj} / \text{Cop} + \text{Neg} + NP / \text{Cop} + \text{Neg} + \text{Adj}.
\end{align*}
\]

(The slash ‘/’ is used to present several rules that have the same nonterminal on the left; thus there are 12 rules given above.) The set \( N_1 \) of nonterminal symbols consists of \( S \), \( NP \), \( VP \), \( UQ \), \( EQ \), \( NQ \), \( N \), \( TV \), \( IV \), \( Cop \), \( Adj \), \( Neg \), each with its nearly classical grammatical meaning — e.g., \( TV \) = transitive verb, \( IV \) = intransitive verb, \( Cop = \text{copula} \), \( UQ \rightarrow All \), \( EQ \rightarrow Some \), and \( NQ \rightarrow No. \) I assume as known the standard definitions of one string of \( V^* \) being \( G \)-derivable from another, the concept of a derivation tree of \( G \), and the language \( L(G) \) generated by \( G \). (For a detailed treatment of these concepts, see Hopcroft & Ullman, 1969.)

We move from syntax to semantics in two stages. First the grammar \( G \) is extended to a potentially denoting grammar by assigning at least one set-theoretical function to each production rule of \( G \). Thus, in the example of Figure 2, the set-theoretical function assigned to the single nonlexical rule is intersection. We may show these functions in general by using brackets to show the denotation of a nonterminal. In the case of Figure 2,
The second stage is the characterization of model structures. In the general theory of model-theoretic semantics for context-free languages, I use the concept of a hierarchy $\mathcal{H}(D)$ of sets built up from a given nonempty domain $D$ by closure under union, subset, and power set 'operations'. A model structure for a given grammar $G$ with terminal vocabulary $V_T$ is a pair $<D,v>$ where $D$ is a nonempty set and $v$ is a partial function from $V_T$ to $\mathcal{H}(D)$. Explicit details are to be found in Suppes (1973a).

Extension of the concept of semantic tree to transformational grammars is in principle conceptually straightforward, but formally complex. Some of the problems are discussed in Suppes (1973b).

### 3. Extended Relation Algebras

Extended relation algebras enter as highly restricted forms of the hierarchy $\mathcal{H}(D)$ of a model structure $<D,v>$. The natural preliminary definition is that of a nonempty family $\mathcal{I}$ of sets closed under the Boolean operations of complementation and union, and the relation operations of converse and relative product, and the image $R^A$ of a set $A$ under the relation $R$. In familiar notation these are just the following conditions: *

- **E1.** If $A \in \mathcal{I}$ then $\neg A \in \mathcal{I}$;
- **E2.** If $A,B \in \mathcal{I}$ then $A \cup B \in \mathcal{I}$;
- **E3.** If $A \in \mathcal{I}$ then $\neg A \in \mathcal{I}$;
- **E4.** If $A,B \in \mathcal{I}$ then $A \circ B \in \mathcal{I}$;
- **E5.** If $A,B \in \mathcal{I}$ then $A''B \in \mathcal{I}$.

The properties of all these operations are discussed in some detail in Suppes (1960). If $D$ is a domain, then $\mathcal{I}$ is a subset of the power set of $D \times D$. For the classical syllogism we need only that $\mathcal{I}$ is an algebra of sets, i.e., is a subset of the power set of $D$ and satisfies E1 and E2. The conditions E1-E5 define a structure that from a purely logical or mathematical standpoint is somewhat unusual, because of E5, but as we saw in the analysis of *John loves Mary* in Figure 4, the image of a set under a

*The image notation, $R^A$, is misprinted later at several points. Hopefully it should be obvious that only one notation is intended.
relation is the natural denotation of a VP node that has a transitive verb as a descendant. More generally, the conditions of closure are required by the semantics of standard S-V-O sentences, except possibly for C4, the closure condition on the relative product, which does not seem to be needed very often in the semantic analysis of common sentences. One possible use is in the semantic analysis of adverbs that intensify a property or relation.

To account for all nodes of semantic trees that denote — in particular, to take account of the values T and F of the Frege function, we may enlarge \( \mathcal{F} \) to \( \mathcal{F} = \mathcal{F} \cup \{T, F\} \) with \( T, F \in \mathcal{F} \) and \( T \neq F \). However, for the algebraic viewpoint of this paper, it is preferable to make the semantic function at the root of a tree not the Frege function used in Figure 4, but a relational function, i.e., a function that is a mapping from \( \mathcal{F} \) to \( \mathcal{F} \) in the case of a single argument, from \( \mathcal{F} \times \mathcal{F} \) to \( \mathcal{F} \) when the function has two arguments, etc. Hilbert and Ackermann's (1950) classic algebraic treatment of the syllogism illustrates the method. Generalizing their treatment we have for the first two production rules of \( G_1 \)

\[
\text{Rule} \quad S \rightarrow \text{UQ} + \text{NP} + \text{VP} \\
\quad S \rightarrow \text{EQ} + \text{NP} + \text{VP}
\]

<table>
<thead>
<tr>
<th>Semantic function</th>
</tr>
</thead>
</table>
| \( [S] = \{\begin{array}{l}
-\{\text{NP}\} \cup \{\text{VP}\} \\
\{\text{NP}\} \cap \{\text{VP}\}
\end{array} \} \) |

The functions shown on the right are Boolean functions — the concept is familiar and I shall not give a formal definition. Later I do use the fact that there are 4 Boolean functions of one argument and 16 such functions of two arguments; e.g., in the case of one argument, if \( A \) is an arbitrary set of a Boolean algebra of sets and \( f \) is a Boolean function of one argument, then \( f(A) \) is \( A, \neg A, \emptyset, \text{or} V \), where \( \emptyset \) is the empty set and \( V \) is the universe. The number of extended relational functions of one or two arguments is more complicated to compute and will not be needed in the sequel.

I now turn to the analysis of quantifiers in object position. I begin with existential quantification because of its greater simplicity, the reason for which will soon be evident. In the initial version we shall move outside extended relation algebras.
Clearly, the meaning of the notation “$\bigcup_j D_j$” takes us beyond a family $\mathcal{F}$ of sets satisfying $E_1-E_5$, but the following elementary theorem of set theory brings us back within $\mathcal{F}$.

Theorem 1. For any sets $A$ and $B$,

$$B''A = \bigcup \bigcup_j [a]_j.$$ 

What this theorem shows is that on the basis of the semantic analysis given here the existential quantifier does not, in standard cases, have any force in object position. Thus the semantic tree of All freshman date some juniors is identical except for the branch with $\text{EQ}$ as label off the VP node in Figure 5. I would claim that this fact is reflected in idiomatic English by the omission of the existential quantifier. Thus it is natural to say Some dogs bite people but pedantic to say Some dogs bite some people, as is also true of the pair of sentences about freshmen. I do not consider here those sentences with existential quantifier in object position which, it is often claimed, have at least one reading with the logical force of $(\exists x)(\forall y) \varphi(x,y)$ rather than $(\forall y)(\exists x) \varphi(x,y)$.

The relational analysis of the universal negative quantifier $\text{no}$ goes hand in hand with that of the existential quantifier, just as in the classical case of I and E propositions. This can be seen by constructing the semantic tree for the sentence

Some people eat no vegetables

Figure 6
If we consider as well the sentence *Some people eat some vegetables*,
the VP denotations of *eat some vegetables* and *eat no vegetables* are
set-theoretic complements of each other: \( \exists \)"V and \(- (\exists \)"V) corresponding exactly to a classical E proposition being the con-
tradictory of the corresponding I proposition.

The analysis of universal quantifiers in the object position is
more difficult. Consider the sentence

*Some freshmen date all juniors*

\[
\begin{array}{c}
S: F \cap \cap D''[j] \\
j \in J \\
EQ \\
VP : \cap D''[j] \\
j \in J \\
NP : F \\
TV : D \\
UQ \\
NP : J
\end{array}
\]

Figure 7

In the universal case, we have

\[
\cap D''[j] \subseteq D''J \\
j \in J
\]

but in general not equality. To account for universal quantifiers
in object position we must extend the preliminary definition of
extended relation algebras. As far as I know, the method of ex-
tension given here is new.

There are two ways of making the extension, the first of
which fits in most naturally with conditions E1-E5, and the
other by explicit definition. The first approach requires adding
an additional closure condition.

Let us define for any two sets A and B, with B intuitively
thought of as a relation:

\[
\cap (B, A) = \cap B''[a]. \\
a \in A
\]

The new closure condition is then just:

E6. If \( A, B \in \mathcal{F} \) then \( \cap (B, A) \in \mathcal{F} \).

A second approach is to eliminate the closure conditions
on \( \mathcal{F} \) altogether and to consider all sets in the power set of
D U (D × D), where D is the domain of individuals of the model structure. In both approaches I am not entirely happy with the inclusion of sets that are "mixtures" of relations and sets of individuals, so another alternative that does not really affect the formal developments here is to take all sets in the union of the power set of D and the power set of D × D. I assume this latter choice in what follows, and I now define ∩ (B, A) in terms of operations closed on this family of sets, which I call $E (D)$ —— $E$ for extended relation algebra of sets. Formally,

$$E (D) = \mathcal{P} (D) \cup \mathcal{P} (D \times D),$$

where $\mathcal{P} (D)$ is the power set, i.e., the set of all subsets, of D.

For the definition of ∩ (B, A), we need the concept of restricting a relation to having a given set as domain. The notation $R \mid A$ is standard in set theory; it is the relation derived from R by restricting the domain of R to the set A. In other words, in terms of intersection and the Cartesian product, we have:

$$R \mid A = R \cap (A \times \langle R \rangle).$$

The next step is to define the set $∩ R"|a|$ in terms of operators $a \in A$ closed in $E(D)$. For this set I use, the notation $∩ (R,A)$. For arbitrary sets A, B, and R, $∩ (R,A) = B$ iff

(i) $B \subseteq R"|A$,
(ii) For every C, if $C \neq O$ & $C \subseteq B$ then $(R\mid A)"C = A$,
(iii) For every D, if $D \neq O$ & $D \subseteq R"|A$ & $D \rightarrow B \neq O$ then $(R\mid A)"(D \rightarrow B) \neq A$.

It is easy to show that $∩(R,A)$ has the intended properties. We thus may replace the semantic tree of Figure 7 by:

$$S : F \cap \cap (D, J)$$
$$\quad \text{EQ}$$
$$\quad \text{VP} : \cap (D, J)$$
$$\quad \text{NP} : F \quad \text{TV} : D \quad \text{UQ} \quad \text{NP} : J$$

Figure 8
4. A Theorem about Quantifiers

A context-free grammar $G = (V, N, P, S)$ is unambiguous if and only if every terminal string in $L(G)$ has exactly one derivation tree (with respect to $G$). A context-free grammar $G' = (V', N', P', S')$ is a conservative extension of an unambiguous grammar $G = (V, N, P, S)$ if and only if (i) $V \subseteq V'$, $N \subseteq N'$, and $P \subseteq P'$, (ii) every terminal string in $L(G)$ has exactly one derivation tree with respect to $G'$. The concept of conservative extension is not standard in the literature — in contrast to the definition just given of unambiguous — but is useful for showing why the conventional linguistic treatment of quantifiers leads to an unnecessary escalation of type semantically. A conservative extension of an unambiguous grammar does not open up a new route to avoid semantic escalation of type.

To prove the point at hand, I restrict consideration to a fragment of English for generating plural forms of the standard A, E, I, and O sentences of the classical syllogism. The grammar $G_2$ will be only partially specified, for I shall omit the terminal English nouns and adjectives that we use to form the ordinary premises and conclusions of syllogisms. The nonterminal vocabulary $N_2 \subseteq N_1$, as given earlier, but the set $P_2$ of production rules is, of course, not a subset of $P_1$. (Thus $G_2$ is not a conservative extension of $G_1.$) The following are assumed, plus the unstated but necessary lexical rules for nouns and adjectives. The rules given are meant to approximate the conventional linguistic view of quantifiers as it would be developed for this small fragment of English. The rules of $P_2$, minus the lexical rules, are:

$$
\begin{align*}
S & \rightarrow \quad NP + VP \\
NP & \rightarrow \quad UQ + N/EQ + N/NQ + N \\
VP & \rightarrow \quad \text{Cop + N/Cop + Adj/Cop + Neg + N/Cop + Neg + Adj} \\
UQ & \rightarrow \quad \text{All} \\
EQ & \rightarrow \quad \text{Some} \\
NQ & \rightarrow \quad \text{No} \\
\text{Cop} & \rightarrow \quad \text{are} \\
\text{Neg} & \rightarrow \quad \text{not}
\end{align*}
$$
The semantics of the VP rules is classical and not an issue here. In particular,

\[ [\text{VP}]=[[\text{N}] / [\text{Adj}] / \neg[\text{N}] / \neg[\text{Adj}], \]

respectively, for the four VP rules given above. Note that the copula, Cop, does not denote. In the sequel, I need only the denotation \([\text{VP}]\) of VP. I also assume without comment the usual model-theoretic relational structures for proving validity and invalidity.

A model structure \(<D, v>\) of a grammar \(G\) is Boolean iff for any string \(s\) of \(V^+_F\) for which \(v\) is defined, \(v(s)\) is a subset of \(D\). The model structure is relational iff \(v(s)\) is an element of \(\mathcal{P}(D)\). A potentially denoting context-free grammar \(G\) is Boolean iff for any Boolean model structure \(<D, v>\) of \(G\), every semantic function of \(G\) has its value in \(\mathcal{P}(D)\) whenever its arguments are subsets of \(D\); the grammar \(G\) being relational is defined in similar fashion. Making explicit the Boolean semantic functions that make \(G_1\) Boolean is routine based on the examples given and the details shall not be repeated. A Boolean grammar is semantically correct iff it has a semantically valid model theory of the syllogism. (The term semantically valid as defined here is really too general. It would probably be better to say syllogistically valid.) The theorem I want now to prove is a negative one about grammar \(G_2\). I interpret the theorem as being a semantically based argument against the possibility of having both a standard linguistic parsing of quantifiers and an appropriately simple model theory of the syllogism.

**Theorem 2.** The grammar \(G_1\) for the syllogism is Boolean and semantically correct. In contrast, neither grammar \(G_2\) nor any of its conservative extensions can be both Boolean and semantically correct.

**Proof.** The proof for \(G_1\) is obvious. I therefore give only the negative proof for \(G_2\). To begin with, I shall assume that \(UQ\) and \(EQ\) do not have a denotation. Later it will be apparent that the same argument works when they do denote. Also for the proof it is necessary to consider only two of the three quantifiers and so I omit any consideration of \(NQ\).
In general, for the rule NP → UQ+N we have a semantic function \( f \) such that

\[ [NP] = f([N]), \]

on the assumption already made that UQ does not denote. Similarly for the rule NP → EQ+N we have a semantic function \( g \) such that

\[ [NP] = g([N]). \]

Suppose these semantic functions are Boolean. Then, as remarked earlier, \( f \) is one of four possible functions, which we may show for an arbitrary set \( A \) of the universe \( V : f(A) \) is \( A, \neg A, V, \) or \( O \). Similarly for \( g \).

Next we suppose there is a Boolean semantic function \( \varphi \) for the top-level rule \( S \rightarrow NP + VP \). We must then have for the universal quantifier

1. \( \varphi([NP],[VP]) = \varphi(f([N]),[VP]) = \neg[N] \cup [VP] \)

and for EQ

2. \( \varphi([NP],[VP]) = \varphi(g([N]),[VP]) = [N] \cap [VP]. \)

Since there are 4 possible functions for \( f \) and 4 for \( g \), we must consider 16 possibilities and show in each case that \( \varphi \) cannot both be Boolean and evaluate \( A \) and I propositions correctly. Put another way, we must show that the Boolean functional equations (1) and (2) do not have a simultaneous solution. It will suffice to consider one case to illustrate the method of argument. Let \( f(A) = A \) and \( g(A) = \neg A \). Then, to cover the UQ needs, we must have

\[ \varphi([N],[VP]) = \neg[N] \cup [VP], \]

but then for the I propositions we will have

\[ \varphi(g([N]),[VP]) = \varphi(\neg[N], [VP]) = [N] \cup [VP], \]

which clearly leads to semantically incorrect evaluations, for we should have \( [N] \cap [VP]. \)

If, on the other hand, UQ and EQ are permitted to denote, \( f([UQ], [N]) \) can be one of 16 Boolean functions and so can \( g([EQ], [N]) \) for a total of 256 possibilities, or, also, we have UQ.
denoting and \( EQ \) not, for 64 possibilities, and still another 64 if we reverse matters so that \( EQ \) denotes and \( UQ \) does not. All the same, it is straightforward but tedious to show that in none of these additional 384 cases can \( \varphi \) both be a proper Boolean function and evaluate \( A \) and \( I \) propositions correctly. The point is that the denotations \([UQ]\) and \([EQ]\) cannot be made to do any real work. Thus we conclude that \( G_2 \) cannot be both Boolean and semantically correct. This conclusion also clearly holds for its conservative extensions.

In looking back over this proof it might be thought that the linguistic approach to quantifiers might be saved by rules like

\[
S \rightarrow NP' + VP \\
NP' \rightarrow UQ + NPEQ + NP,
\]

and so quantifiers are once again classified as parts of noun phrases. But it is easy to show that this syntactic move cannot be endowed with appropriate Boolean functions (for the case of \( A, I, E, \) and \( O \) propositions).

The imposition of formal semantic constraints on the kinds of grammars that can be regarded as acceptable or efficient is, as yet, fairly unexplored territory. Indeed, the restriction to extended relation algebras in this paper has not really been justified by an explicit theoretical argument. Broadly speaking, the arguments are the typical ones for seeking an algebraic formulation of a theory, especially when possible a quantifier-free algebraic version. In the case of language, the more detailed argument must surely be in terms of developing algorithms or semi-algorithms for handling certain linguistic structures that are frequently used.

Smith and Rawson (in press) develop a procedural semantics for natural language that provides an attractive alternative that would be compatible with the standard linguistic treatment of quantifiers and yet not lead to an escalation of type. Roughly speaking, they use partial functions and conditional expressions rather than higher types of sets to provide the appropriate semantic encoding. Their approach works well for many features of natural language, but the model theory is not yet ex-
licitly worked out, and so it is not yet clear how the standard theory of validity and logical consequence is to be developed within their framework. As it is developed, it will be of some interest to compare the computational efficiency of their approach to the algebraic one outlined in this paper.

Stanford University.

REFERENCES

Tarski, A., On the calculus of relations, in Journal of Symbolic Logic, 1941, 6, 73-89.