JOHN N. MARTIN

SOME FORMAL PROPERTIES OF INDIRECT SEMANTICS [1]

The ideas of context free and logistic grammars are defined within the theory of inductive sets, and necessary and sufficient conditions are established for homomorphic translation from the first sort of structure into the second. The method is of interest because of its recent use in the indirect semantic interpretation of context free grammars by translating them into intensional logic. Examples of such interpretations found in the literature fail to meet the required conditions for successful homomorphism.

1. Introduction

In this paper we characterize the conditions under which a context free grammar can be given a model theoretic semantics indirectly by means of a homomorphic translation into a logistic syntax like that of intensional logic. Since logistic structures are as a matter of course homomorphic to semantic structures, the characterization problem reduces to that of stating the conditions for homomorphic translation from a context free grammar into a logistic grammar. [2] The general method of the paper consists of bringing to bear standard definitions and results from the theory of inductive sets, formal grammar, and formal semantics. Since the background theory is well-known, proofs will be provided only for the final results.

In order to compare formal and logistic grammars, a governing cover theory of inductive sets is employed. Being a branch of set theory, it is general enough for the purpose without forcing either theory into an unfam-

---

1 The author would like to thank John S. Schlipf for some helpful comments and corrections and the University of Cincinnati Research Council and Taft Fund which helped support this research.

2 For a general statement of the homomorphic features of semantic interpretation see the definition of 'meaning assignment for L' in Montague (1970), § 3, p. 227 in Thomason (1974).

iliar idiom. Since we shall be considering at most denumerable sets, we appeal only to naive set theory.

A second major assumption of the study is the sense of translation employed in the idea of an indirect interpretation. Translation is understood here in the relatively strict sense of homomorphism from one syntactic structure to another. The choice is justified in part by the resulting characterization. In a well-defined set of cases indirect interpretation is possible under conditions of homomorphism. Moreover homomorphic translation is the intended sense in the literature. The recent work by Gerald Gazdar and others in phrase structure grammar includes as a major feature of its analysis an indirect semantic interpretation. Paired with each type of basic expression of a context free grammar is a category of expressions in intensional logic, and paired with each production rule of the context free grammar is a syntactic rule of intensional logic. A translation is then defined as a homomorphism induced by mapping basic expressions into the corresponding category. Since the syntax of intensional logic has a well-defined model theory, a semantic interpretation of the context free grammar is defined as the composition function composed by pairing with each expression of the context free grammar a model theoretic interpretation of its translation. The process is very much like the indirect interpretation of disambiguated languages in Montague's theory of universal grammar. It differs only in having a context free grammar as the source language where Montague has a disambiguated logistic syntax. As in Montague's theory the device of using intensional logic as a mediating level between the source language and its semantic interpretation is dispensible in principle. If the indirect interpretation is possible, then in principle so is a direct mapping of expressions from the source language into semantic structure.[3]

Indirect translation as a technique of semantic interpretation is certainly not new in linguistics. In transformational grammar the process of transforming surface forms from deep structures may be viewed as a kind of

---

3 For indirect semantics in phrase structure grammar see Gerald Gazdar (1981) and (1982); and Gerald Gazdar et al. (1985). On Montague's indirect semantics see Montague (1970) and David R. Dowty et al. (1981), pp. 178–181 and 254–265. Strictly speaking, Montague's syntax is not required to be a concatenation structure on symbols as stipulated below. We make the narrower stipulation because it is the more common practice among writers on the logistic method and because the extra generality is irrelevant to the characterization problem at hand. The particular examples of languages discussed by Montague, both target and source, count as logistic on the definitions given below.

translation, and so likewise may the derivations from underlying logical forms in generative semantics. Similarly in the philosophical literature the reformulation of ordinary language expressions into canonical notation or logical form is a common device. But these notions of translation differ from the one at hand. The frameworks in which they are used are not algebraic and the translation procedure cannot be construed as a homomorphism induced by the translation of basic expressions on parallel syntactic structures. Translation as homomorphism is an idea that hearkens back rather to the semantics of Tarski and his idea of interpreting a formal language by means of a homomorphic translation into a metalanguage. Indeed, the induced homomorphic translation between object language and metalanguage, on the one hand, and the induced homomorphic interpretation mapping the object language into semantic structure, on the other, are two ways of describing the same phenomenon.[4] The metalinguistic translation of an object language sentence is the metalinguistic statement entailed by Tarski's definition of truth which describes the truth conditions of the object language sentence. What Montague and the phrase structure grammarians may be said to do is to use three structures where Tarski uses two. They employ Tarski's general strategy of interpreting one algebraic structure by mapping it into another, only they do it twice, mapping a source language into a target language and then the target language into semantic structure or, to put it Tarski's way, into parts of the metalanguage. The theoretical advantage of a Tarski-style semantics lies in the succinct and uniform determination of a semantic interpretation for any expression of the language, and this paper may be viewed as exploring some of the formal properties of indirect semantics of this sort.

A third assumption of the study concerns the mode of reformulating context free grammars before mapping them into logistic structures. A derivation in context free grammar consists of a sequence that starts with the sentence symbol S and contains progressively longer strings each of which is derived from the preceding member of the sequence by the expansion of one of its parts in accordance with a production rule. Well-formed expressions in logistic syntax, on the other hand, are derived by tree constructions that start with basic expressions as leaves and take as higher nodes any complex expression which has as its immediate parts the expressions on the immediately preceding nodes, in accordance with one of the rules for well-formed expressions. Thus the linear derivations of a context free grammar must be transformed into tree-like constructions before it can be mapped, homomor-

4 For a clear exposition of this aspect of Tarski's theory see Hartry Field (1972).

phically or otherwise, into a logistic syntax. The usual method of reformulation, a version of which is used here, is to recast the linear derivations as trees that exhibit the relations of constituent structure. One way of doing so is suggested by McCawley and employed by Gazdar in his indirect phrase structure semantics. The proposal is to construe the production rules of context free grammar as node admissibility conditions on constituent structure trees. McCawley's idea and Gazdar's use of it are open to various precise mathematical formulations, and the definitions employed here are designed to facilitate representation in logistic syntax.[5]

One of the causes of the failure in general for context free grammars to homomorphically map into logistic structures is the existence among the former of various kinds of lexical and syntactic ambiguity precluded by definition from the latter. There is yet a more radical reformulation of context free grammars that we shall not pursue here but which does eliminate these ambiguities. On this reformulation a context free grammar is transformed into a formal grammar that has the constituent structure trees of the original grammar as elements and relations on these trees as structural operations. Ambiguities are eliminated because there is a unique tree for each syntactic derivation in the original grammar. Montague employes a similar idea to disambiguate a fragment of English as a preliminary to his indirect semantics, and Wall eliminates the ambiguities of context free grammars in a similar way.[6]

One of the conclusions of our study is that indirect semantics for context free grammars are non-trivial. The particular grammars discussed by Gazdar are a case in point. Though we shall not detail the grammars here, we shall discuss why they fail to meet several of the required conditions for homomorphic translation into intensional logic.

---

5 See McCawley (1973), and Gazdar (1982), esp. pp. 137–140.

6 See Montague (1973) in Thomason (1974), p. 263, and the discussion in Dowty et al. (1981), pp. 254–260, esp. p. 256, and Wall (1972), pp. 214–221.

2. The Theory of Inductive Sets [7]

In the definitions below we make use of the standard notation of set theory.

Definition An inductive set relative to a universe U is any $\langle A, B, F \rangle$ such that

(1) B is a series (possibly infinite) $B_1, \ldots, B_n, \ldots$ of subsets of U;

(2) F is a series (possibly infinite but usually finite) $f_1, \ldots, f_n, \ldots$ of functions of various finite degrees on U;

(3) A is the least set such that all sets in B are included in A and A is closed under the operations in F, or:

$$A = \cap \{C \mid (1) \text{ for any } B_i, \text{ if } B_i \text{ is in } B, \text{ then } B_i \subseteq C; \text{ and } (2) \text{ for any } x_1, \ldots, x_n \in C, \text{ any } f \in F \text{ and } \langle x_1, \ldots, x_n \rangle \in DF, f(x_1, \ldots, x_n) \in C\}.$$

Definition A set C is said to be indexed by a sequence s iff s maps the whole set I^+ of positive integers onto C (in which case s is called an infinite sequence) or s maps some finite subset $\{1, \ldots, n\}$ of I^+ onto C (in which case f is a finite sequence.)

Clearly a sequence is a set of pairs consisting of a positive integer and an element of C. We shall have occasion to speak both of the pair begining with i in s, i.e. $\langle i, s(i) \rangle$, and of the value paired in s with i, i.e. s(i). When the context makes it clear which is meant, we shall refer to either $\langle i, s(i) \rangle$ or s(i) by s_i, and when there is possibility of confusion we shall refer to s(i) in the usual way as s_i, but to $\langle i, s(i) \rangle$ by $\langle s_i \rangle$. It is also customary to refer to the indexing function s by $\langle s_1, \ldots, s_n, \ldots \rangle$ if s is infinite or by $\langle s_1, \ldots, s_n \rangle$ if s is finite.

Definition A sequence s indexing a subset of a universe U is said to be a construction sequence of e relative to a family B of subsets of U, and a set F of functions on U and a function g on s is said to analyze s iff

(1) s_1 is e;

[7] The account of inductive set, construction, the use of constructions sequences to define substitution, and the idea of a logistic grammar may be found in Haskell B. Curry (1963). For a more general account of inductive sets see Yiannis N. Moschovakis (1974), and for a recent discussion of the variety of ideas of construction see Charles McCarthy (1983).

(2) for any $\langle s_i \rangle$ either

(i) there is some B_j in B such that $s_i \in B_j$ and $g(\langle s_i \rangle) = B_j$,

or

(ii) there are some integers k, ..., m and some f in F such that $i < k, \ldots, m$, $f(s_k, \ldots, s_m) = s_i$, and $g(\langle s_i \rangle) = \langle f, k, \ldots, m \rangle$.

For simplicity we shall write $g(s_i)$ for $g(\langle s_i \rangle)$.

This definition is quite general and allows for possibilities that would not normally occur in syntactic constructions or logical proofs. Basic sets may overlap, constructions may be infinitely long[8], items may be generated from themselves, and the same item may have many quite different construction sequences.

Theorem (Induction) If $\langle A, B, F \rangle$ is an inductive set, and

(i) (Basic Step) every B_i in B is a subset of C, and

(ii) (Inductive Step) for any $x_1, \ldots, x_n$ and f, if $f \in F$, $\langle x_1, \ldots, x_n \rangle \in Df$ and $x_1, \ldots, x_n \in C$, then $f(x_1, \ldots, x_n)$ C,

then $A \subseteq C$.

Intuitively, it is not important which order the arguments of a generating function are listed in so long as all the arguments of a function appear earlier in the sequence. Nor does it matter which of two functions is applied first in cases in which the outputs of the functions are not nested inside one another. Thus, we shall call two construction sequences equivalent if they differ only in the order of listing earlier items in the construction or in the order of the generation of non-nested parts.

Definition A construction sequence $s = \langle s_1, \ldots, s_n, \ldots \rangle$ with analysis g on $\langle A, B, F \rangle$ is equivalent to a construction sequence $s' = \langle s'_1, \ldots, s'_n, \ldots \rangle$ with analysis g' on $\langle A, B, F \rangle$ iff there is a $1-1$ function P from $\{1, \ldots, n, \ldots\}$ onto $\{1, \ldots, n, \ldots\}$ (called a permutation) such that for any $\langle s_i \rangle$ of s,

(1) $s_i = s'_p(i)$; and

(2) $g(s_i) = g'(s'_{p(i)})$.

---

8 It is perhaps more usual to require by definition that construction sequences terminate. Thus it is common to find definitions of construction sequence which differ from ours in that they require that the sequence be finite and that e be identical to s_n rather than to s_1 as we have it. Nothing we shall say about the theory of constructive sets turns on the possibility of infinite construction sequences. We prefer the more general definition because it is adequate to the theory and because it simpifies some of our applications of the general ideas to the particular sorts of constructions we shall encounter in the theory of formal grammar.

The second fashion in which constructions are displayed is by means of tree diagrams. The item being constructed is assigned the root node of the tree, and other items of the inductive set are paired with other nodes in such a way that the items paired with the nodes immediately above a node would, if fed into a construction operation, yield the item assigneded the node below. Further the tips of the tree (the "leaves") are all assigned basic elements.

Definition A tree is a structure $\langle T, \leq \rangle$ such that

(1) $\leq$ is a partial ordering of elements of T (it is a reflexive, transitive, and antisymmetric subset of T^2);

(2) there is a unique maximal element t^* of T (called the root) (i.e., t^* is the unique element of T such that for any t in T, $t \leq t^*$);

(3) for any t in T, there is a unique $\langle t_1, \ldots, t_n \rangle$ such that $t_1 = t$, $t_n = t^*$, and for each t_i, $t_i \ll t_{i+1}$, where the relation $\ll$ of immediate predecessor is defined as follows: for any x and y of T, $x \ll y$ iff [$x \neq y$, $x \leq y$, and for any z of T, if $x \leq z$ and $z \leq y$, then either $x = z$ or $y = z$].

The finite sequence defined in clause (3) is said to be the subbranch ending with t. The tree is said to be finitary if T is finite, and to be finitely branching if every node has only a finite number of immediate successors. In general a finitely branching tree need not be finitary, and every branch of a tree may be finitely long yet the tree not be finitely branching nor finitary. A node t is minimal (is a leaf) if it has no $\leq$-successors, i.e., for any t' of T, if $t' \leq t$, then $t' = t$.

We now explain how to represent the construction of an element of an inductive set by a tree. The element constructed is assigned to the tree's root, and assigned to the immediate predecessors of any node are the elements used to generate it by a generating function. To the leaves of the tree are assigned basic elements. The annotation of the tree consists of identifying for each non-basic node the function used to generate it and the nodes, in the right order, to which are assigned the arguments used in the generation. If a node is basic the annotation cites one of the basic sets in which can be found the element assigned to it.

Definition A tree $\langle s, \leq \rangle$ is a construction tree for an element e of a universe U relative to a family B of subsets of U and set F of functions on U, and g annotates $\langle s, \leq \rangle$ iff

(1) s is some possibly infinite sequence indexing some subset of U;

(2) e is s_n such that $\langle s_n \rangle$ is the maximal element of $\langle s, \leq \rangle$;

(3) for any $\langle s_i \rangle$ of s, either

(i) $\langle s_i \rangle$ is minimal and s_i is in some B_j of B, and g(s, i) is one of these (usually there is only one); or

(ii) $\langle s_i \rangle$ is not minimal, and there are some integers k, ..., m and some f in F such that $\langle s_k \rangle, \ldots, \langle s_m \rangle$ are all $\leq$-immediate successors of $\langle s_i \rangle$, $f(s_k, \ldots, s_m) = s_i$, and $g(s, i) = \langle f, k, \ldots, m \rangle$.

Theorem If $\langle A, B, F \rangle$ is an inductive set, then A is:

$$\{x \mid \text{there is a finite construction tree for } x \text{ relative to B and F}\}.$$

Again as in the case of construction sequences, some trees represent essentially the same construction and differ only in unimportant variations in labelling. In the case of trees the order of arguments in a generation and the order of application of generating functions is fixed. But what can vary is the numerical subscript used in identifying a node. The uniform changing of the numerical index for items would in the technical sense generate a new tree, but intuitively the tree would continue to describe the same constructions.

Definition A construction tree $\langle s, \leq \rangle$ relative to B and F on U with annotation g is equivalent to a construction tree $\langle s', \leq' \rangle$ relative to $\langle A, B, F \rangle$ and annotation g′ iff there is a permutation P on $\{1, \ldots, n, \ldots\}$ such that for any $\langle s_i \rangle$ and $\langle s_j \rangle$ of s,

(1) $s_1 = s'_{p(1)}$;

(2) $\langle s_i \rangle \leq \langle s_j \rangle$ iff $\langle s'_{p(i)} \rangle \leq \langle s'_{p(j)} \rangle$;

(3) $g(s_i) = g'(s'_{p(i)})$.

Definition A construction sequences s relative to B and F on U with analysis g is equivalent in a direct sense to a construction tree $\langle s, \leq \rangle$ relative to $\langle A, B, F \rangle$ with annotation g′ iff $g = g'$.

Definition A construction sequence s relative to B and F on U with analysis g is equivalent (in a wide sense) to a constructions tree $\langle s, \leq \rangle$ relative to $\langle A, B, F \rangle$ with annotation g′ iff there is some construction sequence s″ on B and F with analysis g″ and construction tree $\langle s'', \leq \rangle$ on B and F with annotation g″ such that the sequences s and s″ are equivalent, the sequence s″ is equivalent in the direct sense to the tree $\langle s'', \leq'' \rangle$, and the trees $\langle s'', \leq'' \rangle$ and $\langle s, \leq \rangle$ are equivalent.

Like construction sequences, construction trees are defined in a general way that permits possibilities usually excluded from syntactic constructions. These include overlapping basic sets, items generated from themselves (in terms of orderings, some $s_i = s_j$ and $\langle s_i \rangle \leq \langle s_j \rangle$), infinitely long branches, and multiple non-equivalent constructions for the same item.

In an important variety of cases, the domains and ranges of the generating functions form a partitioning of expressions into non-overlapping expression categories. In such grammars if a generating rule is defined for one expression of a category it is defined for all, and the ranges of formation functions are proper parts of these categories. Grammars in logic usually have this property.

Let us single out the set of all i-th arguments of a function f and call it the i-th subdomain of f: $D^i f = \{x \mid \text{for some } y_1, \ldots, y_i, \ldots, y_n, \langle y_1, \ldots, y_i, \ldots, y_n \rangle \in Df \text{ and } x = y_i\}$. In interesting cases these subdomains partitions the inductive set. So that we may identify this and other important properties for discussion, we list some formal definitions.

Definition An inductive set $\langle A, B, F \rangle$ relative to a universe U is:

(1) syntactic iff U is Σ (the set of expressions or symbols) and each f in F is definable in terms of concatenation and set theory (Σ and concatenation are explained below);

(2) finitary iff every construction tree of every element in A is finitary (has a finite number of nodes);

(3) non-circular iff there is no construction tree s for any e in A such that for some distinct i and j, $s_i = s_j = e$ and $\langle s_i \rangle \leq \langle s_j \rangle$;

(4) monotectonic iff each f in F is $1 \rightarrow 1$ and no item is in the range of more than one function in F.

(5) lexically unambiguous iff no element is a member of more than one basic category;

(6) categorized iff any two subdomains of any one or two functions in F are either disjoint or identical, and the range of any function in F is a subset of some subdomain of some function in F.

Theorem If an inductive set is monotectonic, then the construction trees of any element of the set are all equivalent.

Theorem If an inductive set is monotectonic, then all the construction sequences of any element of the set are equivalent.

At this point it is possible to define several very useful syntactic notions like substitution that though intuitively clear require for their formal definitions the concepts just introduced. As has been observed, the same expression may be used more than once in the construction of a longer expression. Multiple use is even possible in non-circular trees if the same item is used to construct different collateral parts of a whole. We therefore define an instance or occurrence of an item e in a construction sequence or tree s as any $\langle s_i \rangle$ such that $s_i = e$.

A second concept that is straightforwardly defined in terms of construction sequences and trees is the grammatical relation of part to whole. It is common to call any expression that enters into the construction of a longer expression its "part". Thus any item which by iterated application of the construction rules together with other arguments leads to a member of the inductive set is a part of the member so constructed. We may then define an element s_i to be part of another s_j relative to a construction tree for an inductive set iff $\langle s_i \rangle \leq \langle s_j \rangle$. Thus the ordering $\leq$ on the tree determines the part-whole relation. Moreover, if the inductive set is monotectonic, we know that all the construction trees of an item have the same ordering relation, so that relative to a monotectonic set we may say s_i is a part of s_j iff relative to any tree, $\langle s_i \rangle \leq \langle s_j \rangle$. Obviously it is possible to give an equivalent definition of "part-whole" using construction sequences rather than trees.

A third idea that is easily definable in terms of constructions is substitution. For various purposes both in grammar and logic we want to substitute one expression for another in the grammatical constructions of a whole. For the process to be well-defined, two assumptions must be met. The first is that there is a unique grammatical construction for the whole in question or, in other words, that the inductive set of well-formed-expressions is monotectonic. If not, there would not necessarily be a single result of substituting one part for another but rather various different results relative to the various possible constructions for the whole. Secondly, the process presupposes that the part being substituted is of the appropriate grammatical type, i. e. that the generating function is defined for it. One way to insure that this assumption is satisfied is to require that the grammar be categorized and that the part being substituted is of the same category as the part it is replacing. The result of replacing the one by the other is then definable as the result of the construction that results from the replacement. In the definition below we use the concept of tree rather than sequence though sequences may be used equally well.

First we define the useful notion of substituting y for z in a construction tree for x. The result is a tree, but proving as much requires appeal to the definition of a tree.

Definition Let $\langle A, B, F\rangle$ be a categorized monotectonic inductive set with elements x, y, and z; let y and z be of the same category; and let $T = \langle s, \leq\rangle$ be a construction tree for x with annotation g. We define $[T]^y_z$ as $\langle s, \leq'\rangle$ such that

(I) $s' = \langle s'_1, \ldots, s'_n, \ldots\rangle$ such that
(a) if $\langle s_i\rangle$ is minimal and $s_i \neq z$, then $s'_i = s_i$;
(b) if $\langle s_i\rangle$ is minimal and $s_i = z$, then $s'_i = y$;
(c) if $\langle s_i\rangle$ is not minimal and $g(s_i) = \langle f, k, \ldots, m\rangle$, then $s'_i = f(s'_k, \ldots, s'_m)$;

(2) for any $\langle s'_i\rangle$, $\langle s'_j\rangle$ of s', $\langle s'_i\rangle \leq' \langle s'_j\rangle$ iff $\langle s_i\rangle \leq \langle s_j\rangle$; and we define a function g' on s' as follows: $g'(s'_i) = \langle f, k, \ldots, m\rangle$ iff $g(s_i) = \langle f, k, \ldots, m\rangle$.

Theorem $[T]^y_z$ meets the conditions for being a construction tree for its maximal element, and g' meets the conditions for being an annotation.

Definition If $\langle A, B, F\rangle$ is a categorized monotectonic inductive set with elements x, y, and z, and if y and z are of the same category, we define the result of substituting y for z in x (written briefly $[x]^y_z$ as the maximal element of $[T]^y_z$ such that T is some construction tree for x.

Exactly what is a constructive set is a matter of wide debate, but mathematicians and logicians would agree that our notion of inductive set would not qualify as a plausible analysis of the idea of constructivity unless it meets the further condition that all the functions generating the set were effective processes. Some functions, called effective or mechanical processes by mathematicians, are such that for any argument a value is determined by a executing a straightforward finitary process. The idea is important and it is possible to analyze it in different ways. One way to do so is by means of what philosophers would call conceptual analysis or definition by necessary and sufficient conditions. An analysis of this sort would take the form of the biconditional:

f is an effective process iff P.

To meet the standards typical of conceptual analysis the biconditional would also have to have "the ring of analyticity" in the sense that it should be customary either in ordinary language or in technical discussions to explain or define the term 'effective process' as equivalent in usage to whatever conditions are formulated by 'P'. Ideally, the terms used in the statement of these conditions would be well understood, either by being embedded in a developed scientific theory or more likely by being central terms of traditional philosophy which even if controversial are nevertheless very familiar. In our discussion here we shall limit ourselves to a philosophical analysis of this sort, but effective process is perhaps primarily a mathematical idea, and mathematicians have explained it axiomatically in the classical work by Gödel, Markov, Turing, Church, and Post.[9]

Informally, an effective process can be characterized first by examples. Euclid's algorithm for long division and the construction of molecular from atomic sentences in the propositional logic are paradigm cases. These procedures are typical of many in mathematics. Their important properties may be roughly abstracted as follows. They all begin with an input that is readily identifiable as an acceptable argument for which the function is defined. Second, the process consists of performing a series of steps which consists of the manipulation of symbols, often with pencil and paper on a page, that have the property that the outcome of each step is transparent to the manipulator. Finally, after a finite number of these steps the process terminates with an output, and it is clear to the manipulator that it has terminated with this output. Each of these stages is in part epistemic. The manipulator is able to know that the function is defined for an argument, that each step terminates in a particular result, and that the whole process terminates in a final result in a finite amount of time. Moreover this knowledge is of a particularly reliable sort. It derives from the fairly crude manipulation of symbols which has beginning and ending stages that are perfectly evident. When the process is performed with pencil and paper on a page the knowledge in question has a kind of perceptual certainty, and it has the reliability typical of straightforward judgments about the arrangements of quite visible marks on a page. We may informally characterize an effective process then as an epistemically evident calculation that proceeds in a finite number of transparent mechanical steps, from an argument that is evidently acceptable, to a value readily identitifable as the end of the process.

---

9 For an introduction to formal characterizations of "effective process" see Martin Davis (1958), and Hans Hermes (1965).

The second philosophical notion relevant to the characterization of constructions is that of a decidable set. Intuitively, a set is decidable if there is an effective test for determining its members. Moreover, this test must be clear and short. Thus decidability is usually defined in terms of effective process.

Definition If A is a subset of U, then A is decidable relative to U (alternatively, A is recursive in U) iff the characteristic function for A relative to U is definable as an effective process.

Definition An inductive set $\langle A, B, F\rangle$ relative to a universe U is constructive iff

(1) B, F, each set in B, and the domain of each f in F are decidable sets (relative to their respective universes), and

(2) each f in F is an effective process.

Definition A logistic grammar is any structure $\langle A, B, F\rangle$ that is inductive, syntactic, finitary, non-circular, monotectonic, lexically unambiguous, and constucture.

3. Formal Grammar

The definitions of inductive set and construction sequence require that the generating relations be functions. It is interesting to note, however, than none of the basic properties of inductive sets depend on this assumption. We may, for example, weaken the definition of inductive set by substituting for clauses (2) and (3) the following:

(2′) F is a set of finitary relations (i.e. for each R in F, there is an n such that $R \subseteq V^n$), and

(3′) for any $x_1, \ldots, x_n, x_m$ and f, if $f \in F$, $x_1, \ldots, x_n \in C$ and $\langle x_1, \ldots, x_n, x_m\rangle \in f$, then $x_m \in C$.

Likewise in the definition of construction sequence the functional condition $f(s_k, \ldots, s_m) = s_i$ may be replaced by the weaker relational requirement $\langle s_k, \ldots, s_m, s_i\rangle \in f$. Let us use the term weak to refer to these amended notions of inductive set and construction relation, and to any notions defined in terms of these notions rather than their stronger originals. The theorems previously stated using the strong notions continue to hold using the weak reformulations. In particular, induction holds for weak inductive sets, and there is a finite weak construction sequence for each element of a weak inductive set.

The philosophical charactericationthat we have used of an effective process is that of an epistemically transparent method for proceeding in a finite number of steps from any starting point to a unique end. On the basis of this definition, it would seem perfectly reasonable to abstract away from the requirement that a unique end be associated with any given starting point. The new notion is that of a method that proceeds from any starting point in a finite number of steps to one of various acceptable ends. At each step it is epistemically transparent whether a prescribed method was applied correctly and what the outcome of the step is. What is no longer required is that each step yield a unique result.

Let us recall moreover the general fact that any finitary relation can be partitioned into functional subsets. Since there is no defining limitation on the number of generating relations in an inductive set, the set of relations of a weak inductive set may be partitioned into functional subsets. A weak inductive set is thus equivalent to some strong inductive set. If any generating relation was effective in the sense proposed here, its functional parts would be also. Conversely, the union of any set of n-place effective functions would seems to produce an effective n + 1-place effective relation in the new sense. Formal grammars are constructions that make use of this relational type of effective process.[10]

We assume an undefined set Σ of symbols, a primitive binary operation $^\frown$ of concatination on Σ, an empty symbol e in Σ, and the set Σ^* of finite strings of elements of Σ. Σ^* may be defined as the universe A of the inductively defined set $\langle A, \Sigma, ^\frown \rangle$.). We also assume that these ideas conform to the following syntactic restrictions:[11]

---

10 For the original statement of the theory of formal grammar see Noam Chomsky (1959) and (1965) in R. Duncan Luce et al. (1965). Formal grammar is to be contrasted with grammar in the logistic sense first formulated clearly in Rudolf Carnap [1934], translated in Carnap (1964).

11 For fully explicit accounts of assumptions about symbols and concatenation used in the logistic traditions see W. V. O. Quine (1951)); Alonzo Church, § 07, "The Logistic Method", in Church (1956); and Haskell B. Curry, Chapter II, "Formal Systems", in Curry (1963).

Axiom of Syntactic Metatheory

$\langle Z^*, ^\frown, e \rangle$ is a group with identity element e in the sense that:

(1) for any x, y and z in Σ^*, $(x^\frown y)^\frown z = x^\frown(y^\frown z)$;

(2) for any x in Σ^*, $x^\frown e = e^\frown x = x$.

It is conventional to let upper-case Roman letters stand for arbitrary members of and for lower case Greek letters (e. g. $\varphi, \psi, \xi, \chi, \omega$) to range over Σ^*.

Every grammar is defined relative to two disjoint finite subsets of Σ, the set N of terminal symbols and the set T of terminal symbols. Intuitively, the nonterminal symbols represent entire parts of speech (e.g. nouns and noun phrases, verbs and verb phrases, prepositions, prepositional phrases,) and are sometimes called phrases markers by linguists. Typical examples would be N, V, NP, VP. It should be stressed however that these are really single symbols, not the sets or categories of symbols that a logician might prefer to speak of. The terminal symbols, on the other hand, are to be understood as rather abstract versions of the particular words of the language that when strung together in the right order produce a grammatical sentence. In the general case we allow that the generating relations in a grammar be any binary relation on Σ^*, and call such relations productions. Let us assume that a special designated symbol S (for "sentence") is in N; and by T^* let use mean the set of finite strings of elements of T.

Definition A formal grammar is any $\langle A, S, P, N, T \rangle$ such that

(1) $\langle A, S, P \rangle$ is a weak inductive set;

(2) P is a finite set of productions (binary relations on Σ^*);

(3) N and T are disjoint finite subsets of Σ.

Definition The language or set of sentences L of a formal grammar $\langle A, S, P, N, T \rangle$ is $A \cap T^*$ (i.e. the set of terminal strings productible in the inductive set).

Definition Two formal grammars are productively equivalent iff their sets of sentences are identical.

A construction sequence relative to S and P of a formal grammar is called a derivation of its first element, and a production is called terminal if its range is a subset of T^*.

There are two properties of productions in terms of which varieties of formal grammars are distinguished. The first is a property of lengthening. If all productions of a grammar have the feature that $\langle x, y \rangle \in P_i$ iff the number of symbols in x (its length) is less than the number of symbols in y, then the set of sentences for a language is straightforwardly decidable: for any string ω of length l, we need only review the finite set of derivations of l or fewer steps to see whether ω is the first element in any of them. In some of the cases discussed below the requirement that productions all yield values longer than their arguments is unnecessarily strong. The same effect can be achieved by weaker assumptions. Even in these cases, however, the decidability of sentences will be established by first showing the weaker grammar is productively equivalent to one with lengthening productions.

A second property of some productions is that they may be described as rewriting rules. Every argument of such a production is characterized by some occurrence of a symbol A, and a value for that argument is any result of replacing A in the argument by a characteristic string ω. Some such replacements are context sensitive in the sense that rewriting is allowed only when A falls within some context $\varphi A \psi$. For any such rule every argument contains at least one occurrence of some $\varphi A \psi$, and a value for it is any replacement of $\varphi A \psi$ by $\varphi \omega \psi$ in the argument. For any such production, then, there is a $\varphi A \psi$ such that an argument $\chi \varphi A \psi \xi$ produces a value $\chi \varphi \omega \psi \xi$. (Here for reasons of lengthening A is required to be a simple symbol while ω is allowed to be any string.) A production on the other hand that allows the rewriting of A by ω regardless of its context is called context free. For any such production, there is an A such that an argument $\varphi A \psi$ produces a value $\varphi \omega \psi$. Both sorts of productions are relations rather than functions because more than one occurrence of A may be rewritten in a single argument.

Definition The context sensitive rule $\varphi A \psi \rightarrow \varphi \omega \psi$ is defined as $\{\langle \chi^\frown\varphi^\frown A^\frown\chi^\frown\xi, \chi^\frown\varphi^\frown\omega^\frown\psi^\frown\xi \rangle \mid \chi, \varphi, \omega, \psi, \xi \in \Sigma^* \ \& \ A \in N\}$.

Definition A formal grammar is context sensitive iff each production is some $\varphi A \psi \rightarrow \varphi \omega \psi$ such that $\omega \neq e$.

Definition The context free rule $A \rightarrow \omega$ is defined as $\{\langle \chi^\frown A^\frown\xi, \chi^\frown\omega^\frown\xi \rangle \mid \chi, \omega, \xi \in \Sigma^* \ \& \ A \in N\}$.

By definition, then, a context sensitive rule $\varphi A \psi \rightarrow \varphi \omega \psi$ is context free in the special case that $\varphi = \psi = e$, and these conform to the lengthening condition that $\omega \neq e$. For the purposes of decidability it proves unnecessary to prohibit for context free rules, as we did for context sensitive, the possibility that a symbol may be rewritten by e.

Definition A formal grammar is context free iff all its productions are context free.

Formal grammars generally, and context free and context sensitive varieties specifically, differ from grammars as they are usually defined by logicians. Some of these differences are inessential and can be eliminated by sometimes tedious but quite trivial reformulations producing equivalent results. Other differences are deep and not eliminable in equivalent ways. We shall discuss the merely apparent differences first.

There are no restrictions on the defining conditions of the generating relations of a formal grammar other than that they be finitary relations. In particular there is no requirement that they be functional. Likewise the defining conditions on the relations of context sensitive and context free grammars do not require that they be functional. For example, there might be a context sensitive rule $\varphi A \psi \rightarrow \varphi \omega \psi$ and a string σ such that σ has two occurrences of the substring $\varphi A \psi$, being broken down as both $\chi \varphi A \psi \xi$ and $\chi' \varphi A \psi \xi'$. In that case both $\langle \sigma, \chi \varphi \omega \psi \xi \rangle$ and $\langle \sigma, \chi' \varphi \omega \psi \xi' \rangle$ are in the relation $\varphi A \psi \rightarrow \varphi \omega \psi$. In the special case in which $\varphi = \psi = e$, the rule would be an example of a non-functional context free production. Thus the main varieties of formal grammar, strictly speaking, fall outside the tradition of grammar in logic because they determine a weak rather than a strong inductive set. For a several reasons, however, this difference is not very important.

First of all we have already seen that weak inductive sets have just as much conceptual claim to be classified as constructions as do strong inductive sets. Secondly, formal grammars defined as weak inductive sets are equivalent to certain strong inductive sets, and in special cases these equivalents will be formal grammars themselves.

A strong inductive set productively equivalent to an arbitrary formal grammar is easily defined if only by considering the degenerate case in which the productions of the grammar are partitioned into the unit sets (i. e. sets containing a single element) of the pairs they contain. Each of these singletons will be a degenerate function. In general, however, a strong inductive set productively equivalent to a formal grammar will not itself be a formal grammar. It might, for example, contain an infinite number of generating functions, as the trivial partition just described would if any of the original productions had an infinite domain. The situation is somewhat different for context sensitive grammars. Though it is true that removing non-functional pairs from productions would in general prevent the structure from qualifying as context sensitive, there are methods for insuring that the new structure remains a formal grammar. The method sketched below exploits the fact that as defined there is a finiteness condition on the number of productions of a formal grammar.

Theorem For any context-sensitive grammar there is a productively equivalent formal grammar with functional productions.

The existence of an equivalent does not provide a method for producing one. In the case of context free grammars, however, there are simple ways to pare-down the context free relational rules into productively equivalent functions. Instead of allowing the rule $A \rightarrow \omega$ to assign multiple values to a string containing A, we require that it assign only one of these, and for regularity we stipulate that this is the value which rewrites the left most occurrence of A. Any derivation using the non-functional rule in its original form may then be recast using the new rule, changing the order of its application as needed.

Definition The left most context free production $A \Rightarrow \omega$ is defined as $\{\langle \chi^\frown A^\frown \xi, \chi^\frown \omega^\frown \xi \rangle \mid \chi, \omega, \xi \in \Sigma^*, A \in N$, and A occurrs only once in $\chi^\frown A\}$.

Definition A left-most context free grammar is any formal grammar in which every production is a left-most context free production.

Theorem Every context free grammar is productively equivalent to the left-most context free grammar formed by taking the left-most version of each of its productions.

The discussion shows that both context sensitive and context free grammars might have been defined in terms of strong inductive sets without altering their productive powers, and that in the case of context free grammars the strong equivalent differs in only minor ways from its weak counterpart. Most presentations of formal grammar theory do not attempt to develop it as part of the theory of inductive class or as part of any similar broad account of construction. It is this sort of account we are attempting here. A formal grammar is to count as a construction in the same sense as a traditional construction in logical syntax and proof theory.[12]

---

[12] One account that does attempt this more general view is M. Gross and A. Lentin (1970) which subsumes the context free grammars under a more abstract notion of a strong inductive class known as a combinatory system. In their exposition the authors

The definition of a formal grammar allows for the possibility of circular construction sequences in the sense that a string may be constructed out of itself. When this occurs, the string occurs more than once in the sequence, and the earlier occurrence (with the greater index value) is cited as an argument for a production used to generate either the later occurrence or some other string that is used to generate the later occurrence.[13] Though we know that every element of even a weak inductive set has at least one finite construction sequence, not all its constructions sequences need be finite. In general, there is no positive integer n such that all construction sequences of ω have some domain m such that $m \leq n$. The method of deciding whether ω is a member of the inductive set requires the inspecting of all construction sequences with domains less than some finite bound n, but in this case the method will not work. Indeed formal grammars in general are undecidable. The decision procedure of inspecting all sequences of a given finite bound will, however, work for the special cases of context sensitive and context free grammars. But the idea requires some preliminary regularization.

For any ω in $(N \cup T)^*$, let its length (briefly, $l(\omega)$) be defined as the number of symbols from $N \cup T$ that are concatenated to form ω, and define $(N \cup T)^n$ to be the set of all strings in $(N \cup T)^*$ of length n. Thus, in general $(N \cup T)^*$ is infinite, but $(N \cup T)^n$ is finite because N and T are. The idea is to insure that the inductive set exhibits the following property of finitary boundedness:

> For any string ω of the inductive set and any construction sequence s of ω, if $i < j$, then $l(s_i) > l(s_j)$.

---

simultaneously maintain three propositions: (1) combinatory systems are defined as having functional rules of generation; (2) productions in context free grammar are assumed to be non-functional rewrite rules as in customary in the theory of formal grammar; and (3) context free grammars are defined as being a special sort of combinatory system. What is necessary to make the account consistent, and what no doubt the authors pressume without saying so, is a regularization step which explains how each context free grammar in the strong sense is productively equivalent to one in the weak sense. It is just such a step we are discussing here. Other general accounts of formal grammar do develop the notion as a special case of wider concepts of constructive class, but the wider notions are weak varieties. See, for example, Wall (1972).

13 Circles in this sense are to be distinguished from what linguists call loops defined as the application of a production twice in a construction so that the argument of one application is either the value of the other or constructed from some value of the other. Loops in this sense are merely instances of inductive constructions and do not in general form circles: the strings generated by the iterated application of the rule are not in general the same string.

If the property holds, then all construction sequences of ω are bounded by the finite value $l(\omega)$, and the decision procedure will work.

First, however, circles and the infinite regresses they engender must be eliminated, and in the case of context free grammars so must the empty rule which allows the length of constructs to decrease rather than increase as the method requires.

The empty rule can be eliminated from a context free grammar without altering productive power as follows. If the rules $A \rightarrow e$, $A \rightarrow \omega$, and $B \rightarrow \varphi A \psi$ are rules of the grammar, construct a grammar like the original except that the rule $A \rightarrow e$ is dropped and the rule $B \rightarrow \varphi \omega \psi$ is added. In general there is no productively equivalent way of deleting a rule $A \rightarrow e$ from a formal grammar that has context sensitive rules, and it is for this reason that the empty rule is prohibited by definition in a context sensitive grammar. In our development here we have not excluded the empty rule from context free grammars because for any such grammar there is a productive equivalent without the empty rule. On the definitions we have used, a regularized context free grammar without the empty rule is a special case of a context sensitive grammar.

A simple way to eliminate circular constructions is by means of the more powerful device of eliminating in a productively equivalent manner all non-terminal singulary rules. These are rules of the form $\varphi A \psi \rightarrow \varphi B \psi$ in a context sensitive grammar and rules of the forma $A \rightarrow B$ in a context free grammar. These are replaced respectively by some rules $\varphi A \psi \rightarrow \varphi \omega \psi$ and $A \rightarrow \omega$ in which $l(A) < l(\omega)$.

Theorem The language of any context sensitive or context free grammar is decidable.

There are two properties of formal grammars that cannot be regularized away in equivalent forms and which make them different from grammars as defined by logicians. Formal grammars unlike their counterparts in logic are in general ambiguous and non-categorized. An inductive set is ambiguous in our earlier terminology if it is not monotectonic or, in other words, if there are two or more non-equivalent construction sequences for some element of the set. Context free grammars, and therefore context sensitive and formal grammars generally, fail to be categorized for two reasons: the domains of productions may overlap without coinciding perfectly, and not every element of the inductive set is part of and argument for some production.

4. Translation from Phrase Structure to Logistic Grammar

We now turn to the conditions under which a regularized context free grammar and a logistic grammar are productively equivalent. We shall see that the match between the two is imperfect in both directions. More interestingly perhaps we shall show in some detail how formal grammars are equivalent to inductive sets. These will be sets of strings constructed from, first of all, atomic strings of terminal symbols and, secondly, formation rules on strings of terminal symbols. These structures are much like logistic grammars even though they fail to have some of the structural properties usually required in logic. In particular, we shall see how the usual tree analysis of a syntactic structure generated in a formal grammar is not the construction tree used to derive it, but rather the construction tree of the same string as derived in a productively equivalent inductive set of the logical sort.

Throughout this section we shall mean by a context free grammar a left-most context free grammar that has been regularized in the sense that it does not contain the empty rule, any non-productive rule or symbol, or any circular constructions. We let s range over construction sequences, and for an annotation function g of s, we write $g(s_i) = A \rightarrow \omega$ instead of $g(s_i) = \langle A \rightarrow \omega, i+1\rangle$. For each occurrence i of a non-terminal symbol A in a construction s (i.e., for each node occupied by A) we define a function that translates the occurrence into what we shall call $(A, i, s)\downarrow$, the string of terminal symbols that it is eventually replaced by in the derivation.

Definition Let G be a context free grammar, s a contruction sequence for some element of G, i some integer in the domain of s, and A a nonterminal symbol of G such that for some χ and ξ, $s_i = \chi^\frown A^\frown \xi$ and $g(s_i) = A \rightarrow \omega$. Then the terminal string which rewrites the occurrence at i of A in s (briefly, $(A, i, s)\downarrow$) is defined recursively as follows:

(1) if $\omega \in T$, then $(A, i, s)\downarrow = \omega$;

(2) if $\omega \in N^*$, then $(A, i, s)\downarrow = (B_1, i-1, a)\downarrow^\frown \ldots ^\frown(B_n, i-1, s)\downarrow$ where $\omega = B_1{}^\frown \ldots {}^\frown B_n$.

We now define the logistic equivalent $A \Rightarrow \omega$ for a production $A \rightarrow \omega$ as a function from n-tuples of strings of terminal symbols to strings of terminal symbols. Intuitively, $A \Rightarrow \omega$ generates the terminal string that rewrites the phrase markers A from the rewritten phrase markers that constitute ω.

Definition Let $A \rightarrow \omega$ be a production of a context free grammar G. Then $A \Rightarrow \omega$ (relative to G) is defined as: $\{\langle (B_1, i-1, s)\downarrow, \ldots, (B_n, i-1, s)\downarrow, (A, i, s)\downarrow \rangle \mid$ s is a construction sequence for some element of G, g is the annotation function of s, i is in the domain of s, $g(s_i) = A \rightarrow \omega$, and $\omega = B_1 \frown \ldots \frown B_n\}$.

Theorem $A \Rightarrow \omega$ is a function.

Theorem $A \Rightarrow \omega$ is $1-1$.

Since the property of having $1-1$ generating functions will prove important shortly, we coin a term to describe such structures.

Definition An inductive set is biunique iff all its generating functions are $1-1$.

Theorem $A \Rightarrow \omega$ is finitary.

Theorem $A \Rightarrow \omega$ is syntactic.

Theorem $A \Rightarrow \omega$ is constructive.

In order to use these new operations to build a structure consisting of strings of terminal symbols, we must identify the appropriate basic elements from which to launch the construction. For any non-terminal category A of the context free grammar, we first define the set of all terminal strings that fall under it in any construction.

Definition If $A \in N$, then the terminal extention of A (briefly [A] is defined as
$\{x \mid$ for some construction sequence s and integer i, $x = (A, i, s)\downarrow\}$.

Theorem $R(A \Rightarrow B_1 \frown \ldots \frown B_n) \subseteq [A]$, and for any i from 1 to n, $D^i/(A \Rightarrow B_1 \frown \ldots \frown B_n) \subseteq [B_i]$.

The relevant basic elements of [A], what we shall call its atomic parts, are comprised by that part of [A] generated by terminal rules.

Definition For $A \in N$, the atomic extension of A (briefly [[A]]) is defined as $\{\omega \mid A \rightarrow \omega$ is a terminal rule$\}$.

Theorem $[[A]] \subseteq [A]$.

Theorem $[[A]] = [A] \cap T^*$.

Definition If $G = \langle A, S, P, N, T \rangle$ is a context free grammar, then its terminal reflection (briefly, GT) is defined as the inductive set $\langle A^*, \{A \Rightarrow \omega | A \rightarrow \omega \in P\}, \{[[A]] | A \in N\}\rangle$.

Theorem $A^* = \cup\{[C] | C \in N\}$.

Proof $A^* \subseteq \{[C] | C \in N\}$ because every basic element of A^* is in some atomic part of some non-terminal category of G and because the generating functions $C \rightarrow \omega$ of GT pair n-tuples of elements in category extensions to elements in category extensions. $\cup\{[C] | C \in N\} \subseteq A^*$ because every element ω in $\cup\{[C] | C \in N\}$ is some $(C, i, s)\downarrow$ relative to G. By reference to this s we can construct a construction sequence in GT for ω. End of Proof.

Corollary $[S] \subseteq A^*$

From the perspective of grammatical theory, the terminal reflections of context free grammars have two intersting features. First, the constituent structure trees commonly cited by linguists in the grammatical analysis of terminal strings turn out to be precisely the contruction trees of the reflected structure. Secondly, not in all cases but in a well defined subset of cases the reflected structure will meet all the conditions for counting as a logistic grammar in the sense of the logical tradition.

The reader may have observed that in the discussion of grammar there are two rather different notions of grammatical tree. First, there is the idea of a construction tree from the theory of inductive sets. Since the constructive operations of a formal grammar are all binary relations, its construction tree will not branch and the notion of a construction tree reduces to that of a construction sequence ordered by the trivial relation holding between s_i and s_j just in case $i \leq j$. Thus any derivation is in a literal sense a construction tree. But a second notion of branching tree that has not be rigorously defined here is also used to sketch the grammatical structure of sentences. These are the familiar constituent structure trees from linguistics that have S as their root, other non-terminal symbols as intermediate nodes, and terminal symbols as leaves. Each descent in the tree, moreover, represents the application of a production rule that rewrites a nonterminal symbol by its successors in the tree. The normal practice in expositions of the theory of formal grammar is to sketch a general procedure for constructing a constituent structure tree from any derivation.[14] Our account here will be more general. We will show

---

14 See Gross and Lentin (1970), p. 83 and Wall (1972), p. 214; and also Ronald V. Book (1973), pp. 4–5.

that every construction sequence of a categorial grammar determines a constituent structure tree on strings of terminal symbols and that this tree is precisely the construction tree defined relative to the terminal reflection of the sentence constructed. The notion of constituent structure tree then turns out to be a special case of the concept of construction tree. Constituent structure trees are construction trees used to generate a particular class of structures on strings of terminal symbols. The exact delimitation of this class of structures will concern us next. It should be noted that some grammarians have advocated adopting constituent structure trees as the central idea of syntactic theory and have reinterpreted the productions of formal grammar as rules defining these trees, and Gazdar proposes such a reading for his phrase structure rules. But the linguists offer no mathematical definition of the phrase structure rules under this new interpretation as conditions of the well-formedness of constituent structure trees, and we shall take up this problem shortly.[15]

In the special case in which the terminal reflection GT of a context free grammar G is monotectonic, any sentence ω of G would have a unique construction sequence in GT. But in general GT is not monotectonic (we investigate the conditions under which it is shortly), and the various construction trees for ω in GT must be defined relative to constructions sequences of G. There are numerous ways in which such a sequence can be mapped onto a constituent structure tree in GT, and different details can be found in standard works on formal grammar.

Theorem For every construction sequence s for a sentence ω of a context free grammar G, there is a construction tree for in GT.

Proof We describe in general terms a procedure for defining a construction tree t in GT from any construction sequence s in G. The two will have the same maximal element. We leave to the reader the straightforward proof that t meets the conditions for being a construction tree of ω in GT.

(1) Start by constructing the trivial tree t′ consisting of the single node S_r such that r is the index of the leftmost (minimal) node of s (and is therefore occupied by S).

(2) From any tree t′ with a minimal node A_r construct a new tree t″

15 See McCawley (1973); Wall (1972), pp. 214–218; Gazdar (1982), esp. pp. 137–140.

as follows: Let r^* be the left-most (greatest) element of s such that for some $B_1, \ldots, B_m$ r^* is annotated by the rule $A \rightarrow B_1{}^\frown \ldots {}^\frown B_m$,

(a) replace A_r by $(A, r^*, s)\downarrow_n$ such that n is that number such that either or the following conditions holds: either n is 1 or the immediate predecessor of A_r in t′ has some annotation $\langle A' \Rightarrow B'_k{}^\frown \ldots {}^\frown B_1, k, \ldots, l\rangle$ such that there exists an i such that, A and n are the i-th elements respectively of $B'_k, \ldots, B'_1$ and k, …, l;

(b) if $A \rightarrow B_1{}^\frown \ldots {}^\frown B_m$ is non-terminal,

(i) add to the annotation function g of t′ the new annotation for $(A, r^*, s)\downarrow_n$ as follows: $g((A, r^*, s)\downarrow_n) = \langle A \Rightarrow B_1{}^\frown \ldots {}^\frown B_m, n+1, \ldots, n+m\rangle$ and

(ii) append to $(A, r^*, s)\downarrow_n$ as immediate descendents $B_{1_{r_1}}, \ldots, B_{m_{r_m}}$ such that r_i is the left most (greatest) element of s in which B_i occurrs;

(c) if $A \rightarrow B_1{}^\frown \ldots {}^\frown B_m$ is terminal, let $g((A, r^*, s)\downarrow_n = [[A]]$.

(3) Continue applying step (2) to the newly produced trees as long as it is applicable, i. e. as long as the tree produced has some minimal element A_r such that A is a non-terminal symbol of G. When (2) is no longer applicable, stop.

Is is now a straightforward matter to show that the result of applying the procedure of the proof to a construction sequence s of G yields a tree that meets the conditions for being a construction tree of GT.

It is also true that the construction trees yielded by this procedure capture the idea of a constituent structure tree commonly found in linguistics, but since this latter notion has in the literature no generally accepted formal definition, this coincidence cannot be rigorously proven. We shall return to the issue of the proper analysis of constituent structure shortly, but for the moment let us assume that every derivation of a context free grammar may be recast as a constituent structure tree understood in the sense of a construction tree in the terminal reflection of the grammar. It is interesting to note that something like the converse of this proposition also holds. Peters and Ritchie have shown that a formal grammar for which every sentence has a constituent structure analysis is a context free grammar.[16] Their

16 See P. Stanley Peters, Jr. and Robert W. Ritchie (1972).

sense of constituent structure tree is essentially the same as ours though it is not formulated in terms of terminal reflections of the logistic sort.

We have already seen how a context free grammar determines a productively equivalent inductive algebra on terminal strings. We now explain the conditions under which this algebra meets the quite strict requirements of a syntax in the logical tradition. Some logistic properties hold trivially of terminal reflections because similar properties hold necessarily of context free grammars. These properties, which were enumerated earlier, may be summarized as follows:

Theorem The terminal reflection of any context free grammar is syntactic, constructive, finitary, and biunique.

Moreover, if there were a circle in any construction sequence for a terminal reflection GT of G, we could determine one for G contradicting the requirement that G is regularized.

Theorem The terminal reflection of any context free grammar is not circular.

Some required properties of logistic grammars, however, fail in general for reflections of context free grammars. They are not generally monotectonic, lexically ambiguous, or categorized.

Theorem Some terminal reflection of some categorial grammar is not monotectonic.

We now formulate necessary and sufficient conditions governing the special case in which a terminal reflection is monotectonic.

Definition A categorical grammar is logistically unambiguous iff there is no ω nor distinct rules of the grammar $A \rightarrow B_1{}^\frown\ldots{}^\frown B_n$ and $A' \rightarrow B'_1{}^\frown\ldots{}^\frown B'_m$ such that ω is a terminal string and an element of both [A] and [A'].

Theorem A context free grammar is logistically unambiguous iff its terminal reflection is monotectonic.

Proof If Part. Suppose GT is not monotectonic. Then there is some string ω, there are some rules $A \Rightarrow B_1{}^\frown\ldots{}^\frown B_n$ and $A' \Rightarrow B'_1{}^\frown\ldots{}^\frown B'_m$, there are some construction sequences s and s', and there are integers i and j such that:

(1) $\omega = (B_1, i, s)\downarrow{}^\frown\ldots{}^\frown(B_n, i, s)\downarrow = (B'_1, j, s')\downarrow{}^\frown\ldots{}^\frown(B'_m, j, s')\downarrow$; but

(2) $\langle (B_1, i, s)\downarrow, \ldots, (B_n, i, s)\downarrow\rangle \neq \langle (B'_1, j, s')\downarrow, \ldots, (B'_m, j, s')\downarrow\rangle$.

Then the rules $A \rightarrow B_1{}^\frown\ldots{}^\frown B_n$ and $A' \rightarrow B'_1{}^\frown\ldots{}^\frown B'_m$ are distinct, for if they were the same (2) would be false. But by (1), ω is an element of both [A] and [A′], contradicting the assumption that G is logistically unambiguous.

Only-if Part. Suppose that there is a string ω and distinct rules $A \rightarrow B_1{}^\frown\ldots{}^\frown B_n$ and $A' \rightarrow B'_1{}^\frown\ldots{}^\frown B'_m$ of G such that ω is an element of both [A] and [A′]. But then (1) and (2) hold as before, and $A \Rightarrow B_1{}^\frown\ldots{}^\frown B_n$ applied to either the argument $\langle (B_1, i, s)\downarrow, \ldots, (B_n, i, s)\downarrow\rangle$ or the argument $\langle (B'_1, j, s')\downarrow, \ldots, (B'_m, j, s')\downarrow\rangle$ yields the value ω. But since by (2) these are distinct arguments, GT is not monotectonic. End of Proof.

According to the strict definition of "lexical ambiguity" as we have defined it in the theory of inductive sets, every formal grammar is trivially lexically unambiguous because a grammar has only the single basic element S. But linguists too speak of lexical ambiguity and have a different notion in mind. A grammar is lexically unambiguous in the linguist's sense if its terminal elements fall under a single category.

Definition A context free grammar is terminally unambiguous iff for any nonterminal symbols A and B of the grammar, if $[[A]] \cap [[B]] = \varnothing$, then $[[A]] = [[B]]$.

Since the various categories [[A]] and [[B]] are exactly the basic categories of a grammar's terminal reflection, it follows immediately that terminal ambiguity in a context free grammar corresponds exactly to lexical ambiguity in its reflection.

Theorem A context free grammar is terminally ambiguous iff its terminal reflection is lexically ambiguous.

The requirement common in logic that grammar be categorized is not in general met by terminal reflections. Both the defining conditions of a categorized grammar may fail: the subdomains of its generating functions may overlap imperfectly and a generating function may produce a value that is not in turn defined for some other function.

Theorem There is some context free grammar such that its terminal reflection is not categorized.

Theorem There is a context free grammar in which the range of some rule $A \Rightarrow \omega$ is not included as a subset of the subdomain of any rule $B \Rightarrow \omega$.

Definition A context free grammar G is categorizable iff

(1) $\{[A] \mid A \in N\}$ partitions the inductive set defined by its terminal reflection GT, and

(2) if ω is any string produced in GT, then there is some rule $A \rightarrow B_1{}^\frown\ldots{}^\frown B_n$ of G, some construction sequence s of G, and some integers i and j such that $\omega = (B_j, i, s)\downarrow$.

Theorem A context free grammar is categorizable iff its terminal reflection is categorized.

Proof Clearly (2) holds iff the range of any $A \Rightarrow B_1{}^\frown\ldots{}^\frown B_n$ is a subdomain of some $A' \Rightarrow B_1'{}^\frown\ldots{}^\frown B_m'$. Likewise if $\{[A] \mid A \in N\}$ partitions the inductive set defined by GT, then the subdomains of any $A \Rightarrow B_1{}^\frown\ldots{}^\frown B_n$ and $A' \Rightarrow B_1'{}^\frown\ldots{}^\frown B_m'$ are either disjoint or identical. Further, if these subdomains are either disjoint or identical, $\{[A] \mid A \in N\}$ partitions GT because by definition $x \in [A]$ only if x is in the subdomain of some generating function of GT. End of Proof.

We may now summarize the conditions under which a context free grammar may be reformulated logistically.

Theorem A categorized context free grammar is lexically and terminally unambiguous iff its terminal reflection is a logistic grammar.

Having set forth the conditions under which a context free grammar determines a logistic grammar, it is natural to inquire about the determination in the reverse direction. Do logistic grammars translate in every case to productively equivalent context free grammars? They fail to do so in general but only because they are defined in a very generous way so as (1) to allow for an infinite number of basic expressions or rules, and (2) to generate expressions that are not necessarily part of some sentence. But in the special case in which the logistic grammar is restricted to finitary resources and sentential parts, it is a straightforward matter to show the grammar's set of sentences coincides with that of a context free grammar.

Theorem Assume L is a logistic grammar with a finite number of basic expressions and generating functions, and that there is some subdomain S (called the set of sentences of L) of some generating function of L which is such that every element generated by L is part of some element of S (i. e. for any element e of L, there is some construction sequence s and some element of P of S such that $P = s_1$ and $e = s_n$, for some $n > 1$). Then S is identical to the set of sentences of some context free grammar.

5. Conclusions

There are two conclusions of some broad conceptual interest that follow from our discussion here. The first concerns the intuitive content of the notion of constituent structure. We have seen that constituent structure analysis is essentially logistic. It consists of displaying construction trees within inductive structures on strings of terminal symbols. These structures do not in general meet all the defining conditions for syntax in logic. Specifically, they fail to be monotectonic and categorized. The logistic properties they do satisfy are impressive and suggest a general definition of a constituent structure grammar.

Definition An algebra is a constituent structure grammar iff it is inductive, syntactic, finitary, non-circular, constructive, and biunique.

The definition is broad enough to capture all the traditional constituent structure analyses in linguistics because it embraces the terminal reflections of all context free grammars.

There may be some worry that it is too broad in that it counts as a constituent structure analyses those of logistic grammars that are not productively equivalent to any context free grammars, e. g. logistic grammars with an infinite number of basic expressions. But all logistic grammars are well behaved from the point of view of construction by constituent parts because they are monotectonic and categorized. For any expression of a logistic grammar there is a construction tree that explains how the expression is constructed from a finite number of basic elements in a finite number of applications of formation rules. There seems to be no motivation for not calling these trees constituent structure trees. One conclusion of our discussion then is that the identification by some linguists of constituent structure analysis with context free grammars is too narrow. Peters and Ritchie are misleading, not in their proof that all constituent structure analyses are essentially context free grammars, but in their definition of constituent structure. The idea should be defined broadly enough to include grammars from the logistic tradition.

A second conclusion is that the technique of providing model theoretic semantical interpretations for the recent phrase structure grammars fails. It does so because the context free grammars the technique is typically applied to fail to have the required structural properties that allow for terminal reflections of a logistic sort. Specifically, the technique presupposes an intensional logic with a precisely defined semantic structure of the model theoretic sort. Thus much of the theory is not problematic. But the theory also presupposes a mapping of the sentences of the context free grammar into those of the syntax of intensional logic, and the mapping seems to be a structure preserving homomorphism. This mapping is not rigorously defined nor is the relevant constituent structure understanding of the context free grammar. Moreover, it seems that any attempt to make these ideas precise must fail because the syntax of intensional logic is logistic. It is categorized, and lexically and terminally unambiguous. But neither of these properties holds of the context free grammar. The grammars proposed are logistically and terminally unambiguous, and they are not categorized. Their productively equivalent constituent structures reformulated as terminal reflections, which employ a precise version of the constituent structure understandings of phrase structure rules as stipulated by Gazdar, are thus not monotectonic and are lexically ambiguous and noncategorized. They are therefore not logistic structures and could not be homomorphic to the syntax of any intensional logic. Nor would any reformulation of the context free grammar as a structure of constituent structure trees, as mentioned in the introduction, suffice to insure homomorphism because though the resulting grammar would be unambiguous it would not in general be categorized.

The desired semantic parallelism between phrases structure grammars and model theoretic constructions may indeed be realizable by means other than Gazdar's technique. What seems to be required first of all is a precise statement of the context free grammars reformulated as constituent structure grammars, perhaps by use of some notion like our terminal reflection. Then two options are open. First, the context free grammars reformulated as consituent structure grammars could be interpreted indirectly through a suitably liberal notion of intensional syntax defined so as be a constituent structure grammar but not logistic in the full sense. But rewriting intensional logic so that it is not a logistic grammar is a major undertaking, and it would be necessary to prove that a homomorphic mapping existed from any reformulated context free grammar to the new sort of syntax for intensional logic. A second alternative would be to abandon any attempt at the kind of short cutting translation envisoned by Gazdar and to interprete the reformulated context free grammars directly by mappings onto a semantic structure. It is not structural limitations on the semantic structures that cause the problem. In the usual definitions of model theoretic semantic structures, there are in general no restrictions requiring the mirroring of logistic properties like that of being categorized, monotectonic, or lexically unambiguous, and hence there is no reason in principle to question whether constituent structure grammars that lack these properties can be homomorphic to semantic structures.

REFERENCES

BOOK, R. V. (1973), Topics in Formal Language Theory. In Alfred V. Aho, ed., Currents in the Theory of Computing. Englewood Cliffs, N. J.: Prentice Hall, 1973.

CURRY, H. B. (1963), Foundations of Mathematical Logic. N. Y.: McGraw-Hill.

CHOMSKY, N. (1959), On Certain Formal Properties of Grammars. Information and Control 2: 137–167.

CHOMSKY, N. (1965), Formal Properties of Grammars. In R. Duncan Luce et al., eds., Readings in Mathematical Psychology, II. N. Y.: Wiley, 1965.

CARNAP, R. (1934), The Logical Syntax of Language. English edition London: Routledge, 1964.

CHURCH, A. (1956), Introduction to Mathematical Logic, I. Princeton, Princeton University Press.

DAVIS, M. (1958), Computability and Unsolvability. N. Y.: McGraw-Hill.

DOWTY, D. R. et al. (1981), Introduction to Montague Semantics. Dordrecht:, Reidel.

FIELD, H. (1972), Tarski's Theory of Truth. Journal of Philosophy 69: 347–375.

GAZDAR, G. (1981), On Syntactic Categories. In H. C. Longuet-Higgins et al., eds., The Psychological Mechanisms of Language. London: The Royal Society, 1981.

GAZDAR, G. (1982), Phrase Structure Grammar. In P. Jacobson and G. K. Pullum, eds., The Nature of Syntactic Representation. Dordrecht: Reidel, 1982.

GAZDAR, G. et al., Generalized Phrase Structure Grammar. Oxford: Blackwell, 1985.

GROSS, M. and LENTIN, A. (1970), Introduction to Formal Grammar. London: Allen & Unwin.

HERMES, H. (1965), Enumerability, Decidability, Computability. N. Y.: Springer-Verlag.

MCCARTHY, CH. (1983), Intuitionism. Journal of Philosophical Logic 12: 105–149.

MCCAWLEY, J. D. (1968), Concerning the Base Component of Transformational Grammar. Foundations of Language 4: 243–269.

MOSCHOVAKIS, Y. N. (1974), Elementary Induction on Abstract Structures. Amsterdam: North-Holland.

MONTAGUE, R. (1970a), Universal Grammar. In Richmond H. Thomason, ed., Formal Philosophy. New Haven: Yale University Press, 1974.

MONTAGUE, R. (1970b), The Proper Treatment of Quantification in Ordinary English. In Richmond H. Thomason, ed., Formal Philosophy. New Haven: Yale University Press, 1974.

PETERS, P. S. and RITCHIE, R. W. (1972), Context Sensitive Immediate Constituent Analysis: Context-Free Languages Revisited. Mathematical Systems Theory 6: 324–333.

QUINE, W. V. O. (1940), Mathematical Logic. Revised edition N. Y.: Harper & Row, 1951.

WALL, R. (1972), Introduction to Mathematical Linguistics. Englewood Cliffs, N. J.: Prentice-Hall.