INFORMATION AND CONTROL 4, 353-358 (1961)

A Note on Mathematical Induction on Phrase Structure Grammars

ROBERT W. FLOYD

Armour Research Foundation of Illinois Institute of Technology, Chicago 16, Illinois

Two rules of derivation are exhibited and shown to yield valid metalinguistic theorems concerning phrase structure grammars (type 2 or context-free grammars, in Chomsky's notation).

DEFINITIONS. The notion of a *character* or letter is taken as undefined. Characters will be represented by small Greek letters. An *alphabet* or vocabulary is a finite collection of distinct characters. For convenience, an alphabet will be assumed to be an ordered set $\{\alpha_1, \alpha_2, \cdots, \alpha_n\}$; this assumption is not essential. A *string* is a finite nonempty sequence of characters chosen from a given alphabet. In particular, each character is itself a string. Strings will be represented by small Latin letters. A grammatical *category* is a (possibly infinite or empty) set of strings. Categories will be represented by Latin capitals. A *language* is a finite set of categories, of which one may optionally be singled out as the category of sentences or *well formed formulas*.

The length of a string x, written $l(x)$, is the number of characters in the string. For any string x and character α, $l(x) \geqq 1$ and $l(\alpha) = 1$. The *concatenation* xy of two strings x and y is the string of length $l(x) + l(y)$ whose first $l(x)$ characters are those of x in the order in which they appear in x, and whose next $l(y)$ characters are those of y in order. Concatenation is associative; $(xy)z = x(yz)$. If there exists a string y such that $xy = z$, then x is a *head* of z; similarly, if $yx = z$, x is a *tail* of z. In either case, $l(x) < l(z)$.

The *union* or disjunction of two categories A and B, written $A \cup B$, is the set $\{x \mid x \in A \vee x \in B\}$. The *product* AB is the set of strings of the form xy, where $x \in A$ and $y \in B$. A *primary category* $\{\alpha\}$ is that whose sole element is the string α consisting of a single character.

A *phrase structure language* (PSL) is a set of categories $A_i (1 \leqq i \leqq m)$

each of which is defined by

$$A_i = \{\alpha_{j_i}\} \tag{1}$$

or

$$A_i = A_{k_i} A_{l_i} \tag{2}$$

or

$$A_i = A_{k_i} \cup A_{l_i} \tag{3}$$

Not every such set of definitions serves to define a language, in the sense of determining unambiguously whether a given string belongs to a given category. Consider the set of definitions

$$A = B \cup C$$

$$B = C \cup A$$

$$C = A \cup B$$

Clearly an arbitrary string may be consistently assumed to belong to each of A, B, and C, or to none of them. The incompleteness of the set of definitions may be traced to the presence of a cycle of disjunctive definitions. A *phrase structure grammar* (PSG) is a set of definitions of the three forms given above, where if A_i is defined by $A_i = A_{k_i} \cup A_{l_i}$ then $k_i < i$ and $l_i < i$. This restriction serves to eliminate disjunctive cycles without diminishing the set of definable languages (proof is omitted here). Such a grammar is equivalent to a type 2 (or context-free) grammar in Chomsky's notation (Chomsky, 1959) or a simple phrase structure grammar in the notation of Bar-Hillel *et al.* (1960).

Consider a proposition T concerning a PSG whose categories are $A_i (1 \leqq i \leqq m)$, and whose letters are $\alpha_j (1 \leqq j \leqq n)$. Suppose T takes the form $\bigwedge_{i=1}^{m}(x \in A_i \supset P_i\,(x))$, abbreviated $\bigwedge_{i=1}^{m} T_i$.

THEOREM 1. *If for each* A_i *defined by* $A_i = \{\alpha_{j_i}\}$, *there is a proof of* $P_i\,(\alpha_{j_i})$; *and for each* A_i *defined by* $A_i = A_k A_l$, *there is a proof of* $P_k\,(x) \wedge P_l\,(y) \supset P_i\,(xy)$; *and for each* A_i *defined by* $A_i = A_k \cup A_l$, *there is a proof of* $P_k\,(x) \vee P_l\,(x) \supset P_i\,(x)$; *then there is a proof of* $\bigwedge_{i=1}^{m}(x \in A_i \supset P_i\,(x))$, abbreviated $\bigwedge_{i=1}^{m} T_i$ or simply T.

PROOF: T is equivalent to the conjunction

$$(l(x) = 1 \supset T_1) \wedge (l(x) = 1 \supset T_2) \wedge \cdots \wedge (l(x) = 1 \supset T_m) \wedge$$

$$(l(x) = 2 \supset T_1) \wedge (l(x) = 2 \supset T_2) \wedge \cdots \wedge (l(x) = 2 \supset T_m) \wedge$$

$$(l(x) = 3 \supset T_1) \wedge \cdots \text{ etc.,}$$

or$\Lambda_{p=1}^{\infty}(l(x) = q + 1 \supset T_{r+1})$, where q and r are the quotient and remainder respectively of $(p-1)/m$. Let us designate T^{π} as $\Lambda_{p=1}^{\pi}(l(x) = q+1 \supset T_{r+1})$. We shall show that $T^{\pi} \supset T^{\pi+1}$. Since T^{0} is vacuously true, T^{π} is true for all π, and therefore T is true.

Assume the truth of T^{π}. Only the last term of $T^{\pi+1}$ remains to be proven. The last term of $T^{\pi+1}$ is $l(x) = Q+1 \supset T_{R+1}$, where Q and R are quotient and remainder respectively of π/m. Three cases must be distinguished.

Case 1

If A_{R+1} is defined by $A_{R+1} = \{\alpha_{j_{R+1}}\}$, there is a proof of $P_{R+1}(\alpha_{j_{R+1}})$, and thus of $x \in A_{R+1} \supset P_{R+1}(x)$ and of $l(x) = Q+1 \supset T_{R+1}$.

Case 2

If A_{R+1} is defined by $A_{R+1} = A_k A_l$, and $x \in A_{R+1}$, $l(x) = Q+1$, then $x = yz$, where $y \in A_k$ and $z \in A_l$, $l(y) < Q+1$ and $l(z) < Q+1$. Now T^{π} implies $P_k(y)$ and $P_l(z)$. Since $P_k(y) \wedge P_l(z) \supset P_{R+1}(yz)$, we may deduce $l(x) = Q+1 \supset (x \in A_{R+1} \supset P_{R+1}(x))$, the last term of $T^{\pi+1}$.

Case 3

If A_{R+1} is defined by $A_{R+1} = A_k \cup A_l$, and $x \in A_{R+1}$, $l(x) = Q+1$, then $x \in A_k$ or $x \in A_l$. Since $k < R+1$ and $l < R+1$, T^{π} implies $P_k(x) \vee P_l(x)$. Since $P_k(x) \vee P_l(x) \supset P_{R+1}(x)$, we may deduce $l(x) = Q+1 \supset (x \in A_{R+1} \supset P_{R+1}(x))$.

No matter which definition scheme is used for A_{R+1}, $T^{\pi} \supset T^{\pi+1}$. Therefore T is provable by mathematical induction.

EXAMPLE. Let Odd(x) and Even(x) stand for the assertions that $l(x)$ is odd or even respectively. Let $P(A_i)$ mean $x \in A_i \supset P(x)$. Define a PSG by $A = \{\alpha\}$, $B = CA$, $C = DA$, and $D = A \cup B$. Then to prove Odd(A) $\wedge$ Odd(B) $\wedge$ Even(C) $\wedge$ Odd(D) requires only proof of Odd(α), Even(x) $\wedge$ Odd(y) $\supset$ Odd(xy), Odd(x) $\wedge$ Odd(y) $\supset$ Even(xy), and Odd(x) $\vee$ Odd(x) $\supset$ Odd(x), each of which is obvious.

By an induction analogous to that of Theorem 1, it is possible to prove assertions concerning the heads and tails of the strings of a category. Certain properties of heads will first be remarked. If $A = \{\alpha_j\}$, then $xy \notin A$. If $A = BC$, and $xy \in A$ then (1) $x \in B$, $y \in C$, or (2) there exist strings u and v such that $x = uv, u \in B, vy \in C, l(vy) < l(xy)$, or (3) there exist strings u and v such that $y = uv$, $xu \in B$, $v \in C$, $l(xu) < l(xy)$. If $A = B \cup C$, and $xy \in A$, then $xy \in B$ or $xy \in C$. Consider a proposition of the form $\Lambda_{i=1}^{m} xy \in A_i \supset P_i (x)$.

Theorem 2. *If for each A_i defined by $A_i = A_k A_l$ there is a proof of $x \in A_k \supset P_i (x)$ and of $u \in A_k \wedge P_l (v) \supset P_i (uv)$ and of $P_k (x) \supset P_\imath (x)$; and if for each A_i defined by $A_\imath = A_k \cup A_l$ there is a proof of $P_k (x) \vee P_l (x) \supset P_i (x)$; then there is a proof of $\Lambda_{i=1}^{m} xy \in A_i \supset P_i (x)$.*

Proof: The proof of Theorem 2 is much like that of Theorem 1, except that the induction is carried out on $l(xy)$ rather than $l(x)$.

For an integer i such that A_i is a primary category, there are no strings x and y such that $xy \in A_i$. It is convenient to choose for $P_\imath (x)$ an identically false proposition, such as $x \neq x$. This simplifies certain steps in the proof procedure; if $P_l (v)$ is false for all v, then $u \in A_k \wedge P_l (v) \supset P_i (uv)$ is true for all u and v, etc.

An analogous procedure, interchanging the roles of x and y, allows proof of propositions of the form $\Lambda_{i=1}^{m} xy \in A_i \supset Q_i (y)$.

Corollary. *If for each A_i defined by $A_i = A_k A_l$ there is a proof of $y \in A_l \supset Q_i (y)$ and of $Q_k (u) \wedge v \in A_l \supset Q_i (uv)$ and of $Q_l (y) \supset Q_i (y)$; and if for each $A_\imath$ defined by $A_\imath = A_k \cup A_l$ there is a proof of $Q_k (y) \vee Q_l (y) \supset Q_\imath (y)$; then there is a proof of $\Lambda_{i=1}^{m} xy \in A_i \supset Q_i (y)$.*

A *linear string function* (LSF) is a function $f(x)$ whose domain is the set of strings over a given alphabet, and whose range is a subset of the real numbers, such that for all strings x and y, $f(xy) = f(x) + f(y)$. The length of a string is a LSF. The LSF's $f_j (x)$ whose values are completely determined by $f_j (\alpha_k) = \delta_{jk}$ form a basis for the vector space of LSF's. If A is a category and $f(x)$ a LSF, we shall write $f(A)$ to mean an arbitrary element of the range of $f(x)$ when x is restricted to be a member of A. To prove for a given PSG that $f(A_i) = c_i$, where c_i is a constant depending only on i, it is necessary and sufficient to prove for each A_i

(1) If $A_\imath = \{\alpha_j\}$, that $f(\alpha_j) = c_i$.

(2) If $A_i = A_k A_l$, that $c_\imath = c_k + c_l$.

(3) If $A_i = A_k \cup A_l$, that $c_\imath = c_k = c_l$.

The set of LSF's which take on constant values for each of several categories A_i is a vector subspace of the set of all LSF's.

EXAMPLE. Define a language L_p by

$$\begin{array}{ll} V = \{\phi_0\} & Q = P \cup U \\ U = \{\phi_1\} & R = QF \\ B = \{\phi_2\} & F = R \cup V \\ P = BF & \end{array}$$

Interpreting ϕ_0 as a variable, ϕ_1 as a unary prefix operator, ϕ_2 as a binary prefix operator, and F as the category of well formed formulas, L_p is the well known Polish prefix notation. The categories P, Q, and R are auxiliary. Defining f as the linear string function with $f(\phi_0) = 1, f(\phi_1) = 0, f(\phi_2) = -1$, it is readily proven that $f(V) = 1, f(U) = 0, f(B) = -1, f(P) = 0, f(Q) = 0, f(R) = 1, f(F) = 1$, using Theorem 1. It may then be proven using Theorem 2 and its corollary that

$$xy \in P \supset (f(x) \leqq -1 \wedge f(y) \geqq 1)$$

$$xy \in Q \supset (f(x) \leqq -1 \wedge f(y) \geqq 1)$$

$$xy \in R \supset (f(x) \leqq 0 \wedge f(y) \geqq 1)$$

$$xy \in F \supset (f(x) \leqq 0 \wedge f(y) \geqq 1)$$

REMARKS. The variants of mathematical induction offered above as mechanisms for reasoning about phrase structure grammars suffer from the defect that the theorem proven must refer explicitly to every category of the language. This unpleasant situation is unavoidable. In most languages of practical interest, whether natural or artificial, there is a complicated interdependence of categories, so that a proof must simultaneously consider properties of each category. If a particular set of categories is independent it may be treated as a sublanguage with a resulting simplification of proofs. A proof procedure of general applicability, however, can not assume this possibility.

Familiarity with the uses and pitfalls of the two proof procedures described allows them to be used in a more informal manner. The language L_p, for example, may be defined informally by $F = \phi_0 \cup \phi_1 F \cup \phi_2 FF$. Inspection then indicates that for any LSF such that $f(F) = 1$ it is necessary that $f(\phi_0) = 1, f(\phi_1) = 0$, and $f(\phi_2) = -1$.

RECEIVED: May 27, 1961

References

Chomsky, N. (1959). On certain formal properties of grammars. *Information and Control* **2**, 137–167; A note on phrase structure grammars. *Information and Control* **2**, 393–395.

Bar-Hillel, Y., Perles, M., and Shamir, E. (1960). On formal properties of simple phrase structure grammars. *Tech. Rept. No. 4*, Applied Logic Branch, Hebrew University of Jerusalem.