JOURNAL OF COMPUTER AND SYSTEM SCIENCES 14, 130–149 (1977)

Hypergrammars: An Extension of Macrogrammars

V. J. RAYWARD-SMITH

School of Computing Studies, University of East Anglia, Norwich, NR4 7TJ, England

Received May 29, 1974; revised September 25, 1975

A new class of generative grammars called hypergrammars is introduced. They are described as a natural extension of Fischer's macrogrammars. Three modes of derivation, inside-out, outside-in, and unrestricted are considered, and the classes of languages so defined are compared with other known classes. It is shown that the outside-in hyperlanguages are the same as the outside-in macrolanguages but that inside-out hyperlanguages are the same as Fischer's quoted languages. Various closure properties are considered as well as generalizations of the original definitions. Three new hierarchies of languages each embedded in the class of quoted languages are discovered. It is claimed that this new approach to Fischer's work is more understandable and also mathematically elegant.

INTRODUCTION

Generative grammars have been of interest to the computer scientist and the mathematician since they were first introduced by Chomsky in 1956 [3]. The languages of type 0, 1, 2, 3 (first classified by Chomsky in 1959 [4]) have been studied in detail resulting in a considerable literature. These classes of languages are denoted by $\mathscr{L}_0$, $\mathscr{L}_1$, $\mathscr{L}_2$, and $\mathscr{L}_3$, respectively. But these are not the only classes of languages generated by grammars; indexed grammars [1], programmed grammars [8], and macrogrammars [5] are but a few of the numerous new forms of generative grammar recently studied. Macrogrammars are probably the most interesting of these; the motivation for the definition of these grammars is a generalization of the concept of a macro as met in programming. In this paper the productions of a macrogrammar are extended to operate on sets of strings instead of just strings. This simple idea clarifies much of Fischer's work, especially in regard to his extension of macrogrammars, namely, quoted grammars.

The new classes of grammars and associated languages are called hypergrammars and hyperlanguages, respectively. As with macrogrammars, there are various modes of derivation, namely IO, OI, and unrestricted. The classes of languages generated by hypergrammars under these various modes are considered in this paper and compared with the classes of languages defined by Fischer. In particular, it is shown that the class of OI hyperlanguages is equal to the class of OI macrolanguages and that the class of IO hyperlanguages is equal to the class of quoted languages. This new approach to Fischer's work results in the discovery of three new hierarchies of languages. The description and proof of these hierarchies is given in Section 6 of the paper.

ISSN 0022-0000

Throughout this paper considerable use is made of the notation and concepts of algebras. The concept of a free algebra is especially important.

DEFINITION 0.1. Let N denote the set of nonnegative integers. A *stratified alphabet* is a set Ω with an associated function $\tau\colon \Omega \to N$ called the *rank* or *arity*. If $\omega \in \Omega$ then the integer $\tau(\omega)$ is called the rank or arity of ω.

DEFINITION 0.2. An *algebra* $\mathscr{A}$ is a pair (A, Ω) where A is a set of elements called the *carrier* of the algebra and Ω is a stratified alphabet of *operator symbols* such that, for each $\omega \in \Omega$, there is an associated operation $\mathscr{A}\omega\colon A^{\tau(\omega)} \to A$. It is common to denote the operation $\mathscr{A}\omega$ simply by ω when it is clear from the context which algebra is under consideration. With operator symbols Ω, $\mathscr{A}$ is often called an Ω-algebra. The carrier of the algebra $\mathscr{A}$ is denoted by $|\mathscr{A}|$.

DEFINITION 0.3. A *subalgebra* of an Ω-algebra $\mathscr{A}$ is an Ω-algebra $\mathscr{B}$ such that

(1) $|\mathscr{B}| \subseteq |\mathscr{A}|$;

(2) $\mathscr{B}\omega(x_1, \ldots, x_{\tau(\omega)}) = \mathscr{A}\omega(x_1, \ldots, x_{\tau(\omega)})$, $\forall\, x_1, \ldots, x_{\tau(\omega)} \in |\mathscr{B}|$.

DEFINITION 0.4. Let $\mathscr{A}$ and $\mathscr{B}$ be two Ω-algebras. A function $h\colon |\mathscr{A}| \to |\mathscr{B}|$ defines a *homomorphism* from $\mathscr{A}$ to $\mathscr{B}$ provided that $\forall\omega \in \Omega$ and $\forall x_1, \ldots, x_{\tau(\omega)} \in |\mathscr{A}|$, $\mathscr{B}\omega(h(x_1), \ldots, h(x_{\tau(\omega)})) = h(\mathscr{A}\omega(x_1, \ldots, x_{\tau(\omega)}))$. If h also has an inverse function $h^{-1}\colon |\mathscr{B}| \to |\mathscr{A}|$ which defines a homomorphism from $\mathscr{B}$ to $\mathscr{A}$ then h is called an *isomorphism* between $\mathscr{A}$ and $\mathscr{B}$. $\mathscr{A}$ and $\mathscr{B}$ are then said to be isomorphic.

DEFINITION 0.5. Let P denote the set of symbols {"(", ")", ",", "}, A any set, and Ω an arbitrary stratified alphabet with arity function τ. $\Omega(A)$, the set of polynomials over Ω, A, is the least set of strings over $(A \cup \Omega \cup P)^*$ such that

(1) $a \in \Omega(A)$, $\forall a \in A$;

(2) if $\omega \in \Omega$ and $t_1, \ldots, t_{\tau(\omega)} \in \Omega(A)$, then the string $\omega(t_1, \ldots, t_{\tau(\omega)}) \in \Omega(A)$.

The *free Ω-algebra generated by A* is the algebra $\mathscr{A} = (\Omega(A), \Omega)$ where the operation $\mathscr{A}\omega$, for $\omega \in \Omega$, is defined by

$$\mathscr{A}\omega(t_1, \ldots, t_{\tau(\omega)}) = \omega(t_1, \ldots, t_{\tau(\omega)}), \text{ for any } t_1, \ldots, t_{\tau(\omega)} \in \Omega(A).$$

The free Ω-algebra generated by A is denoted by $\mathscr{F}_\Omega(A)$.

1. Definition and Examples of Macrogrammars

Before formally defining macrogrammars, consider two examples of such grammars.

Example 1. The macrogrammar G has productions

$$S \to f(a, b, c),$$
$$f(X_1, X_2, X_3) \to f(aX_1, bX_2, cX_3),$$
$$f(X_1, X_2, X_3) \to X_1X_2X_3,$$

where S is the start symbol, f is a function symbol of arity 3, a. b, c are terminal symbols, and X_1, X_2, X_3 are variable symbols. The following is a valid derivation in G:

$$\begin{aligned} S &\Rightarrow f(a, b, c) && \text{(by application of the 1st production),} \\ &\Rightarrow f(aa, bb, cc) && \text{(by application of the 2nd production),} \\ &\Rightarrow f(aaa, bbb, ccc) && \text{(by application of the 2nd production),} \\ &\Rightarrow a^3b^3c^3 && \text{(by application of the 3rd production).} \end{aligned}$$

In fact G will generate $\{a^nb^nc^n \mid n \geqslant 1\}$.

Example 2. The macrogrammar G' has productions

$$S \to f(g),$$
$$f(X) \to XX,$$
$$g \to a,$$
$$g \to b,$$

where S is the start symbol (which is regarded as a specially designated function symbol of arity 0), f is a function symbol of arity 1, and g is a function symbol of arity 0. a, b are terminal symbols and X is the only variable symbol. A valid unrestricted derivation is $S \Rightarrow f(g) \Rightarrow gg \Rightarrow ag \Rightarrow ab$. If no term involving a function symbol can be rewritten until all the arguments of the function symbol are terminal symbols, the derivation is called an inside-out derivation. The derivation of ab from S is not an inside-out derivation because $f(g) \Rightarrow gg$ is not allowed. $S \Rightarrow f(g) \Rightarrow f(a) \Rightarrow aa$ is a valid inside-out derivation. The grammar generates $\{aa, ab, ba, bb\}$ with unrestricted derivation, but only $\{aa, bb\}$ with inside-out derivation.

Definition 1.1. Macrogrammars operate over structured strings called *macroterms*. Macroterms are built up from elements of

(1) a finite set T of *terminals*;

(2) a finite stratified alphabet (F, τ) of *function symbols*, i.e., a finite set F such that

for each $f \in F$ there is a unique nonnegative integer $\tau(f)$, the *arity* of f. The arity of f determines the number of arguments taken by f;

(3) *punctation symbols* "(", ")" and ", ".

The set of macroterms over T, F, τ, $\mathscr{M}(T, F, \tau)$, is the least set of strings over $T \cup F \cup \{$"(", ")", ", "$\}$ such that

(1) the empty string $\epsilon \in \mathscr{M}(T, F, \tau)$;

(2) $\forall a \in T$, $a \in \mathscr{M}(T, F, \tau)$;

(3) if $f \in F$ and $t_1, \ldots, t_{\tau(f)} \in \mathscr{M}(T, F, \tau)$ then $f(t_1, \ldots, t_{\tau(f)}) \in \mathscr{M}(T, F, \tau)$.

Note that writing $f(t_1, \ldots, t_{\tau(f)})$ does not imply that it is necessary that $\tau(f) > 0$. If $f \in F$ is of arity zero, $f(\ )$ is abbreviated to f.

DEFINITION 1.2. Let $t \in \mathscr{M}(T, F, \tau)$. v is a *subterm* of t if $v \in \mathscr{M}(T, F, \tau)$ and v is a substring of t. A subterm v of t is said to occur *at the top level* in t if there exist macroterms t_1, t_2 such that $t = t_1 v t_2$, i.e., if v does not appear in t within the argument list of some function symbol.

DEFINITION 1.3. A *macrogrammar structure* (MGS) is a 6-tuple (N, T, F, τ, P, S) where

(1) N is a finite set of *variables*;

(2) T is a finite set of *terminals*;

(3) (F, τ) is a stratified alphabet of *function symbols*;

(4) P is a finite set of (*macro*) *productions* of the form $f(X_1, \ldots, X_{\tau(f)}) \to \gamma$ where $X_i \in N$, $i = 1, \ldots, \tau(f)$, and $\gamma \in \mathscr{M}(T \cup \{X_1, \ldots, X_{\tau(f)}\}, F, \tau)$;

(5) $S \in F$ is the *start symbol* where $\tau(S) = 0$.

DEFINITION 1.4. Let $t_1, t_2 \in \mathscr{M}(T', F, \tau)$. Let $G = (N, T, F, \tau, P, S)$ be an MGS such that $T' \cap F = \varnothing$ and $T' \supset T$. We define

(1) $t_1 \Rightarrow_G^{\text{unr}} t_2$, t_1 *directly derives* t_2 *by an unrestricted step*, if

(i) t_1 contains a subterm $f(v_1, \ldots, v_{\tau(f)})$ where $f \in F$ and $v_1, \ldots, v_{\tau(f)} \in \mathscr{M}(T', F, \tau)$;

(ii) P contains the rule $f(X_1, \ldots, X_{\tau(f)}) \to \gamma$;

(iii) t_2 is obtained from t_1 by replacing a single occurrence of the subterm $f(v_1, \ldots, v_{\tau(f)})$ described in (i) by γ' where γ' is obtained from γ by substituting the macroterms $v_1, \ldots, v_{\tau(f)}$ for the corresponding occurrences of $X_1, \ldots, X_{\tau(f)}$ in γ.

(2) $t_1 \Rightarrow_G^{\text{IO}} t_2$, t_1 *directly derives* t_2 *by an inside-out step*, if

(i) $t_1 \Rightarrow_G^{\text{unr}} t_2$;

(ii) all the arguments of the rewritten function symbol are elements of T'^*.

(3) $t_1 \Rightarrow_G^{\text{OI}} t_2$, t_1 *directly derives* t_2 *by an outside-in step*, if

(i) $t_1 \Rightarrow_G^{\text{unr}} t_2$;

(ii) the subterm of t_1 which is rewritten occurs at the top level in t_1.

$\Rightarrow_G^{\text{unr}}$, $\Rightarrow_G^{\text{IO}}$, $\Rightarrow_G^{\text{OI}}$ are written as $\Rightarrow^{\text{unr}}$, $\Rightarrow^{\text{IO}}$, $\Rightarrow^{\text{OI}}$ when it is clear which MGS is under consideration. $\Rightarrow^*$, $\Rightarrow^{\text{IO}*}$, and $\Rightarrow^{\text{OI}*}$ are defined as the reflexive, transitive closure of $\Rightarrow^{\text{unr}}$, $\Rightarrow^{\text{IO}}$, and $\Rightarrow^{\text{OI}}$, respectively.

DEFINITION 1.5. A *macrogrammar* $\mathbf{G}$ is a pair $(G, \Rightarrow_{\mathbf{G}})$ where G is an MGS and $\Rightarrow_{\mathbf{G}}$ is a mode of derivation, either $\Rightarrow_G^{\text{unr}}$, $\Rightarrow_G^{\text{IO}}$, or $\Rightarrow_G^{\text{OI}}$. An *unrestricted macrogrammar* (UMG) is a macrogrammar $(G, \Rightarrow_G^{\text{unr}})$. Similarly $(G, \Rightarrow_G^{\text{IO}})$ is an *IO macrogrammar* (IOMG) and $(G, \Rightarrow_G^{\text{OI}})$ is an *OI macrogrammar* (OIMG).

DEFINITION 1.6. Let $\mathbf{G} = ((N, T, F, \tau, P, S), \Rightarrow_{\mathbf{G}})$ be any macrogrammar. A term $t \in \mathscr{M}(T, F, \tau)$ is a *sentential form* of $\mathbf{G}$ if $S \Rightarrow_{\mathbf{G}}^* t$. The *language generated by the macrogrammar* $\mathbf{G}$, $L(\mathbf{G})$, is the set of sentential forms of $\mathbf{G}$ which are terminal strings.

$$L(\mathbf{G}) = \{w \in T^* \mid S \Rightarrow_{\mathbf{G}}^* w\}.$$

DEFINITION 1.7. The class of languages generated by UMGs is denoted by μ_{unr}. Such languages are called *unrestricted macrolanguages* (UMLs).

The class of languages generated by IOMGs is denoted by μ_{IO}. Such languages are called *inside-out macrolanguages* (IOMLs).

The class of languages generated by OIMGs is denoted by μ_{OI}. Such languages are called *outside-in macrolanguages* (OIMLs).

Fischer [5] proves

THEOREM 1.1. (1) *For any* MGS, G, $L((G, \Rightarrow^{\text{unr}})) = L((G, \Rightarrow^{\text{OI}}))$. *Hence* $\mu_{\text{unr}} = \mu_{\text{OI}}$.

(2) $\mathscr{L}_2 \subsetneqq \mu_{\text{IO}} \subsetneqq \mathscr{L}_1$.

(3) $\mathscr{L}_2 \subsetneqq \mu_{\text{OI}} \subsetneqq \mathscr{L}_1$.

(4) μ_{OI} *is precisely the class of indexed languages as defined by Aho* [1].

Fischer also produces Example 3 to show that there is an element of μ_{OI} not in μ_{IO}, and Example 4 to show that there is an element of μ_{IO} not in μ_{OI}. So μ_{IO} and μ_{OI} are incomparable.

EXAMPLE 3. Define a homomorphism $\psi: \{a, b\}^* \to \{a\}^*$ by $\psi(a) = a$, $\psi(b) = \epsilon$. Let $L = \{a^{2^n} \mid n \geqslant 1\}$. $L' = \psi^{-1}(L) = \{x \in \{a, b\}^* \mid \psi(x) \in L\}$. L is generated by the macrogrammar with productions

$$S \to f(a),$$
$$f(X) \to f(XX),$$
$$f(X) \to XX,$$

under any mode of derivation. L' is generated by the OIMG with productions

$$S \to f(g),$$
$$f(X) \to f(XX),$$
$$f(X) \to XX,$$
$$g \to bg,$$
$$g \to gb,$$
$$g \to a.$$

Fischer proves that $L' \notin \mu_{\text{IO}}$.

EXAMPLE 4. The language $\{1^m(c1^m)^{n-1} \mid m \geqslant 1, n = 2^m\}$ is generated by the IOMG with productions

$$S \to f(1),$$
$$f(X) \to g(f(X1)),$$
$$f(X) \to g(X),$$
$$g(X) \to XcX.$$

Fischer shows that this language $\notin \mu_{\text{OI}}$.

THEOREM 1.2. *μ_{OI} (and hence μ_{unr}) are closed under union, concatenation, Kleene closure, intersection with regular sets, homomorphisms, and reversal. However, only μ_{OI} (and hence μ_{unr}) is closed under inverse homomorphisms.*

Proof. [5].

The proofs given by Fischer involve the use of results concerning normal forms for IOMGs and OIMGs. These are also detailed in [5].

2. QUOTED GRAMMARS

Quoted grammars were introduced by Fischer as an extension of IOMGs. New symbols "$\langle$" and "$\rangle$", called *quotes*, were used as a control in the derivation. Let T, F, τ be defined as for macrogrammars.

DEFINITION 2.1. The set of *quoted terms* over T, F, τ, $\mathcal{Q}(T, F, \tau)$, is the least set of strings over $T \cup F \cup \{$"(", ")", ", ", "$\langle$", "$\rangle$"$\}$ such that

(1) $\epsilon \in \mathcal{Q}(T, F, \tau)$;

(2) $\forall a \in T$, $a \in \mathcal{Q}(T, F, \tau)$;

(3) if t_1, $t_2 \in \mathcal{Q}(T, F, \tau)$, then $t_1 t_2 \in \mathcal{Q}(T, F, \tau)$;

(4) if $f \in F$ and $t_1, \ldots, t_{\tau(f)} \in \mathcal{Q}(T, F, \tau)$, then $f(t_1, \ldots, t_{\tau(f)}) \in \mathcal{Q}(T, F, \tau)$;

(5) if $t \in \mathcal{Q}(T, F, \tau)$, then $\langle t \rangle \in \mathcal{Q}(T, F, \tau)$.

DEFINITION 2.2. A *quoted grammar structure* (QGS) is defined as for an MGS except that if $f(X_1, ..., X_{\tau(f)}) \to \gamma$ is a production, then γ may include quotes, since the restriction we impose on γ is that $\gamma \in \mathcal{Q}(T \cup \{X_1, ..., X_{\tau(f)}\}, F, \tau)$.

There are two types of derivation step allowed:

(1) *Quote removal*: This can only occur when quotes occur at the top level, i.e., the quotes do not appear within the argument list of some function symbol. The quotes are removed in matched pairs, e.g.,

$$\langle\langle bf(b)\rangle f(\langle g\rangle)\rangle \Rightarrow \langle bf(b)\rangle f(\langle g\rangle) \Rightarrow bf(b)f(\langle g\rangle).$$

The quotes surrounding g cannot be removed at present since they do not occur at the top level.

(2) *Function rewriting*: A subterm may be transformed by a function rewriting step, as defined in Definition 1.4, only if

(i) it is a term of the form $f(t_1, ..., t_{\tau(f)})$ where each t_i is fully expanded, i.e., the t_i are terminal strings or cannot be further expanded because of quotes. Such terms are called *fully quoted*.

(ii) The subterm is not contained within higher level quotes.

Thus the derivation is inside-out where a quoted term is considered as a terminal and quotes can only be removed at the top level.

EXAMPLE 5. If a quoted grammar has productions

$$\begin{aligned} S &\to hf(b)f(\langle g\rangle), \\ f(X) &\to XcX, \\ g &\to a, \qquad g \to b, \\ h &\to b, \end{aligned}$$

then a valid derivation is

$$\begin{aligned} S &\Rightarrow hf(b)f(\langle g\rangle) \\ &\Rightarrow bf(b)f(\langle g\rangle) \\ &\Rightarrow bbcbf(\langle g\rangle) \\ &\Rightarrow bbcb\langle g\rangle\, c\langle g\rangle \\ &\Rightarrow bbcbgc\langle g\rangle \\ &\Rightarrow bbcbac\langle g\rangle \\ &\Rightarrow bbcbacg \\ &\Rightarrow bbcbacb. \end{aligned}$$

DEFINITION 2.3. The language generated by a quoted grammar **G** (i.e., a QGS together with the above described mode of derivation) is defined by

$$\underset{\sim}{Q}(\mathbf{G}) = \{w \in T^* \mid S \Rightarrow^* w\}.$$

Such a language is called a *quoted language* and the class of all such languages is denoted by $\underset{\sim}{Q}$.

THEOREM 2.1 (Quoted normal form). *For every quoted grammar* **G**, *there is a quoted grammar* **G**′ *which generates the same language and such that every production of* **G**′ *is of one of the following forms*:

(i) $f(X_1, \ldots, X_{\tau(f)}) \to g(h_1(X_1, \ldots, X_{\tau(f)}), \ldots, h_{\tau(g)}(X_1, \ldots, X_{\tau(f)}))$,
(ii) $f(X_1, \ldots, X_{\tau(f)}) \to \eta$, $\eta \in (N \cup T)^*$,
(iii) $f(X_1, \ldots, X_{\tau(f)}) \to \langle g(X_1, \ldots, X_{\tau(f)})\rangle$.

Proof. [5].

THEOREM 2.2. (1) $\underset{\sim}{Q} \supsetneq \mu_{\mathrm{IO}} \cup \mu_{\mathrm{OI}}$.

(2) $\underset{\sim}{Q}$ *is closed under union, concatenation, Kleene closure, substitution, and arbitrary homomorphism.*

Proof. [5, 7].

Closure under intersection with regular sets and under inverse homomorphism remains an open problem. The solution of either of these problems will solve the other [6].

3. HYPERGRAMMARS

In Example 2 the macrogrammar with productions

$$\begin{aligned} S &\to f(g), \\ f(X) &\to XX, \\ g &\to a, \qquad g \to b, \end{aligned}$$

was considered. This example clearly illustrated the difference between IO and OI modes of derivation. Now, if the concepts used so far are extended to sets of strings and not just individual strings, some interesting results are obtained. Consider the productions

$$\begin{aligned} S &\to f(g), \\ f(X) &\to XX, \\ g &\to \{a, b\}. \end{aligned}$$

Under IO or OI derivation, $S \Rightarrow f(g) \overset{*}{\Rightarrow} \{a, b\}\{a, b\}$. If the notion of concatenation of strings is extended to that of concatenation of sets, $\{a, b\}\{a, b\} = \{aa, ba, ab, bb\}$, i.e., the language obtained from Example 2 by OI derivation.

This simple idea is basically the principle behind hypergrammars. All terminals $a \in T$ are considered as singleton sets $\{a\}$; all concatenations are regarded as set concatenation and we also allow the empty set $\varnothing$ and the use of union symbols on the right-hand sides of productions. The modes of derivation for hypergrammars are the same as those for macrogrammars, namely, unr, IO, and OI.

EXAMPLE 6. Consider the two hypergrammars with productions

⟨A⟩	⟨B⟩
$S \to f(g)$	$S \to f(g),$
$f(X) \to XX,$	$f(X) \to XX,$
$g \to \{a\} \cup \{b\},$	$g \to \{a\}, \quad g \to \{b\}.$

The language generated by Example 6A under any mode of derivation is $\{aa, ba, ab, bb\}$. The language generated by Example 6B under IO derivation is $\{aa, bb\}$; under OI derivation it is $\{aa, ba, ab, bb\}$.

It appears that if sets of strings are allowed on the right-hand side, then IO derivation is all that is needed to reflect both the IO derivation and OI derivation used with macrogrammars. In this particular case there is no difference in Example 6A between the languages generated by the various modes of derivation, but there is a difference in Example 6B.

EXAMPLE 7. Consider the hypergrammar with productions

$$S \to f(g),$$
$$f(X) \to f(XX) \cup XX, \quad f(X) \to XX,$$
$$g \to bg \cup gb \cup a, \quad g \to bg, \quad g \to gb, \quad g \to a.$$

Under IO derivation this grammar generates the language $L' = \psi^{-1}(L)$ of Example 3.

Note that the vinculum parentheses "{", "}" have been dropped when writing singleton sets, e.g., a is written for $\{a\}$. This convention will be adopted henceforth. The formal definition of a hypergrammar can now be presented.

DEFINITION 3.1. A *hypergrammar structure* (HGS) is a 6-tuple $G = (N, T, F, \tau, P, S)$, where

(1) N is a finite set of *variables*;

(2) T is a finite set of *terminals*, $\hat{T} = T \cup \{\epsilon, \phi\}$;

(3) (F, τ) is a finite stratified alphabet of *function symbols*;

(4) The operator set Σ is constructed from F by adding to F the two binary operators *union* and *concat*. Then P is a finite set of productions of the form $f(X_1, \ldots, X_{\tau(f)}) \to \gamma$ where γ is an element of the free Σ-algebra generated by $\hat{T} \cup \{X_1, \ldots, X_{\tau(f)}\}$;

(5) $S \in F$ is the start symbol and $\tau(S) = 0$.

DEFINITION 3.2. An element of $\mathscr{F}_\Sigma(\hat{T})$, the free Σ-algebra over $\hat{T}$, is called a *hyperterm*. Let $\Delta = \Sigma \backslash F = \{union, concat\}$, then $\mathscr{F}_\Delta(\hat{T})$ denotes the free Δ-algebra over $\hat{T}$.

Concat (t_1, t_2) is usually written $(t_1 t_2)$ or $(t_1 \cdot t_2)$ and *union* (t_1, t_2) is usually written $(t_1 \cup t_2)$. The parentheses are dropped when no confusion can arise because of their omission.

DEFINITION 3.3. If $v, t \in \mathscr{F}_\Sigma(\hat{T})$ then v is a *subterm* of t if v is a substring of t. An occurrence of a subterm v of t is said to *occur at the top level* in t if that occurrence of v does not appear in t within the argument list of some function symbol in F.

If $f \in F$ occurs as a substring of $t \in \mathscr{F}_\Sigma(\hat{T})$, then the *scope* of that occurrence of f in t is the least subterm of t containing that occurrence of f. Thus in $g(af(a, b), b)$ the scope of f is the string $f(a, b)$.

DEFINITION 3.4. Let $t_1, t_2 \in \mathscr{F}_\Sigma(\hat{T})$ and $G = (N, W, F, \tau, P, S)$ be an HGS such that $T \cap F = \varnothing$ and $T \supset W$. Then

(1) $t_1 \Rightarrow_G^{\text{unr}} t_2$, t_1 *directly derives* t_2 *by an unrestricted step* if

(i) t_1 contains a subterm $f(v_1, \ldots, v_{\tau(f)})$ where $f \in F$ and $v_1, \ldots, v_{\tau(f)} \in \mathscr{F}_\Sigma(\hat{T})$;

(ii) P contains the rule $f(X_1, \ldots, X_{\tau(f)}) \to \gamma$;

(iii) t_2 results from t_1 by replacing a single occurrence of $f(v_1, \ldots, v_{\tau(f)})$ by γ', constructed from γ by replacing each X_i by v_i, $i = 1, \ldots, \tau(f)$.

(2) $t_1 \Rightarrow_G^{\text{IO}} t_2$, t_1 *directly derives* t_2 *by an inside-out step* if

(i) $t_1 \Rightarrow_G^{\text{unr}} t_2$;

(ii) all the arguments of the rewritten function symbol are elements of $\mathscr{F}_\Delta(\hat{T})$.

(3) $t_1 \Rightarrow_G^{\text{OI}} t_2$, t_1 *directly derives* t_2 *by an outside-in step* if

(i) $t_1 \Rightarrow_G^{\text{unr}} t_2$;

(ii) the subterm of t_1 which is rewritten occurs at the top level in t_1.

$\Rightarrow_G^{\text{unr}*}$, $\Rightarrow_G^{\text{IO}*}$, and $\Rightarrow_G^{\text{OI}*}$ are defined as the reflexive, transitive closures of $\Rightarrow_G^{\text{unr}}$, $\Rightarrow_G^{\text{IO}}$, and $\Rightarrow_G^{\text{OI}}$, respectively.

DEFINITION 3.5. A hypergrammar is an HGS together with a defined mode of derivation, either unrestricted, inside-out, or outside-in, as defined above. According to the choice of mode of derivation, the hypergrammar is called an *unrestricted hypergrammar*, an *IO hypergrammar*, or an *OI hypergrammar*.

Now, writing *union* (t_1, t_2) as $(t_1 \cup t_2)$, *concat* (t_1, t_2) as $(t_1 t_2)$ and inserting all parentheses, a hypergrammar is just a macrogrammar with special terminal symbols {"(", ")", "$\cup$", "$\varnothing$"}. The interest in hypergrammars lies in the fact that these symbols have been defined as a convenient way of dealing with sets of strings. Given a hyperterm which involves no function symbols, there is an evaluation map, eval, which produces a set of strings in T^*.

DEFINITION 3.6. 2^{T^*} denotes the Δ-algebra with carrier 2^{T^*} and operations defined by

$$\textit{union}\ (A, B) = A \cup B \text{ (set theoretic union), and}$$

$$\textit{concat}(A, B) = A \cdot B \text{ (set concatenation)}$$

$$= \{ab \mid a \in A, b \in B\}.$$

eval: $\mathscr{F}_\Delta(\hat{T}) \to 2^{T^*}$ is defined as the Δ-algebra homomorphism defined by $\text{eval}(\varnothing) = \varnothing$, the empty set, and $\text{eval}(a) = \{a\}$, $\forall a \in T \cup \{\epsilon\}$.

If x is a sentential form of a hypergrammar $\mathbf{G}$ (with terminal set T) and $x \in \mathscr{F}_\Delta(\hat{T})$, then we say that x is a *terminal term* of the hypergrammar $\mathbf{G}$.

For the rest of this paper, it is assumed that unless otherwise specified all hypergrammars operate over the same terminal set T.

DEFINITION 3.7. The *language generated by a hypergrammar* $\mathbf{G} = (G, \Rightarrow_{\mathbf{G}})$ is defined to be the set of terminal strings $H(\mathbf{G}) = H((G, \Rightarrow_{\mathbf{G}})) = \bigcup \{\text{eval}(x) \mid S \Rightarrow_{\mathbf{G}}^* x \text{ and } x \in \mathscr{F}_\Delta(\hat{T})\}$. The set $\{x \mid S \Rightarrow_{\mathbf{G}}^* x \text{ and } x \in \mathscr{F}_\Delta(\hat{T})\}$ is denoted by $L(\mathbf{G})$. $L(\mathbf{G})$ is a macro-language and is called the unevaluated language generated by $\mathbf{G}$.

Note that $\text{eval}((t_1 \cup t_2) \cup t_3) = \text{eval}(t_1 \cup (t_2 \cup t_3))$ and $\text{eval}((t_1t_2)t_3) = \text{eval}(t_1(t_2t_3))$. Hence $((t_1 \cup t_2) \cup t_3)$ is often written as $(t_1 \cup t_2 \cup t_3)$ and $((t_1t_2)t_3)$ as $(t_1t_2t_3)$ without fear of confusion. In fact, in general, parentheses are omitted where omission can cause no confusion. When parenthese are omitted, a representation of an element of $\mathscr{F}_\Sigma(\hat{T})$ is obtained. In particular, the set of terminal terms consists of elements of $\mathscr{F}_\Delta(\hat{T})$. Dropping parenthese only produces representations of terminal terms.

It is assumed, without loss of generality, that when considering an IO derivation of a terminal term the scope of the rightmost function symbol is always rewritten as the next step in the derivation. Similarly, for the OI derivation of a terminal term, the scope of the leftmost function symbol is always rewritten.

EXAMPLE 8. Consider the hypergrammar $\mathbf{G}$ with productions

$$S \to f(a, b, c),$$

$$f(X_1, X_2, X_3) \to f(aX_1, bX_2, cX_3) \cup X_1X_2X_3,$$

$$f(X_1, X_2, X_3) \to \varnothing.$$

Under any mode of derivation, $\mathbf{G}$ generates the same language $\{a^nb^nc^n \mid n \geqslant 1\}$. This language is well known not to be context free.

DEFINITION 3.8. If there exists a hypergrammar $\mathbf{G}$ generating a language H, then H is called a *hyperlanguage*. The hyperlanguage is called unrestricted, IO, or OI depending upon the mode of derivation of the grammar $\mathbf{G}$.

$\mathscr{H}_{\text{unr}}$, $\mathscr{H}_{\text{IO}}$, $\mathscr{H}_{\text{OI}}$ denote the classes of unrestricted, IO, and OI hyperplanguages, respectively.

DEFINITION 3.9. A hypergrammar $\mathbf{G} = (G, \Rightarrow_{\mathbf{G}})$ is in *standard form* if every production of G is of one of the forms

(1) $f(X_1, \ldots, X_{\tau(f)}) \to g(X_1, \ldots, X_{\tau(f)}) \cup h(X_1, \ldots, X_{\tau(f)})$,

(2) $f(X_1, \ldots, X_{\tau(f)}) \to g(h_1(X_1, \ldots, X_{\tau(f)}), \ldots, h_{\tau(g)}(X_1, \ldots, X_{\tau(f)}))$,

(3) $f(X_1, \ldots, X_{\tau(f)}) \to \eta$, $\eta \in (N \cup T)^*$,

(4) $f(X_1, \ldots, X_{\tau(f)}) \to \varnothing$.

THEOREM 3.1 (Standard Form Theorem). *For every hypergrammar* $\mathbf{G} = (G, \Rightarrow_{\mathbf{G}})$, *there exists a hypergrammar* $\mathbf{G}' = (G', \Rightarrow_{\mathbf{G}})$ *in standard form such that* $H(\mathbf{G}) = H(\mathbf{G}')$.

Proof. [7].

Stronger normal form theorems are available for hypergrammars for any given mode of derivation. These theorems are given in [7] but for this paper the standard form will suffice. One interesting result, though, is that productions involving $\varnothing$ are not necessary, i.e., for any hypergrammar $\mathbf{G} = (G, \Rightarrow_{\mathbf{G}})$ there is a hypergrammar $\mathbf{G}' = (G', \Rightarrow_{\mathbf{G}})$ with no productions involving $\varnothing$, such that $H(\mathbf{G}') = H(\mathbf{G})$.

THEOREM 3.2. $\mathscr{H}_{\text{unr}} = \mathscr{H}_{\text{OI}} = \mu_{\text{OI}}$.

Proof. $\mathscr{H}_{\text{unr}} = \mathscr{H}_{\text{OI}}$: $H \in \mathscr{H}_{\text{unr}}$ iff $H = H((G, \Rightarrow^{\text{unr}}))$ for some hypergrammar $(G, \Rightarrow^{\text{unr}})$ iff $H = \text{eval}(L((G, \Rightarrow^{\text{unr}})))$ iff $H = \text{eval}(L((G, \Rightarrow^{\text{OI}})))$ (Theorem 1.1(1)) iff $H = H((G, \Rightarrow^{\text{OI}}))$ iff $H \in \mathscr{H}_{\text{OI}}$.

$\mathscr{H}_{\text{OI}} = \mu_{\text{OI}}$: It can be shown that if $f(X_1, \ldots, X_{\tau(f)}) \to g(X_1, \ldots, X_{\tau(f)}) \cup h(X_1, \ldots, X_{\tau(f)})$ is a production of an OI hypergrammar G then $H((G, \Rightarrow^{\text{OI}}))$ is unchanged either if this production is removed completely or if this production is replaced by $f(X_1, \ldots, X_{\tau(f)}) \to g(X_1, \ldots, X_{\tau(f)})$ and $f(X_1, \ldots, X_{\tau(f)}) \to h(X_1, \ldots, X_{\tau(f)})$. An easy induction argument can now be used to produce an OI macrogrammar producing the same language as the standard form OI hypergrammar generating $H((G, \Rightarrow^{\text{OI}}))$.

It is interesting to compare IO hyperlanguages with quoted languages. Consider a derivation in a quoted grammar of a terminal string x from a quoted term t. Such a derivation is inside-out except that quoted terms are treated like terminal symbols. If there are no quotes in t then all such IO derivations can be ordered so that the scope of the rightmost function symbol is always that involved in the next expansion. If a subterm v of t is quoted then v is treated like a terminal string until it occurs at the top level; there the quotes are removed and the derivation continues. The one occurrence of v in t can lead to several occurrences of v in later sentential forms. Each of these occurrences may generate different strings of teminals. For example, if a quoted grammar has productions

$$S \to f(\langle g \rangle),$$

$$g \to a,$$

$$g \to b,$$

$$f(X) \to XcXd,$$

then a derivation is $S \Rightarrow f(\langle g\rangle) \Rightarrow \langle g\rangle\, c\langle g\rangle d \overset{*}{\Rightarrow} gcgd$. Each occurrence of g can generate one of a or b. If it was deemed necessary to expand g before f then it would be essential to keep the option of either a or b open. Hence, the IO hypergrammar with productions

$$S \to f(g),$$
$$g \to a \cup b,$$
$$f(X) \to XcXd,$$

is constructed.

In general, when a term v is quoted, the corresponding term in an IO hypergrammar must contribute the same set of terminal strings as generated by v. There is a clear link between "quoting" and "unioning."

Consider the quoted grammar with productions

$$S \to f(\langle g\rangle),$$
$$f(X) \to a, \qquad f(X) \to X.$$

This grammar generates the language $\{a\}$. g itself can generate no terminal string but $f(\langle g\rangle)$ can generate a terminal string by having a derivation in which the quoted g is dropped. In the hypergrammar, the symbol $\varnothing$ acts in a similar way. Its occurrence will enable the derivation to continue but it contributes nothing to the language generated by the grammar. Thus from the example above, the IO hypergrammar with productions

$$S \to f(g),$$
$$g \to \varnothing,$$
$$f(X) \to a, \qquad f(X) \to X$$

is constructed.

The above gives the motivation for the constructions used to prove the following theorem.

THEOREM 3.3. $\mathscr{H}_{\text{IO}} = Q$.

Proof. (1) $\mathscr{H}_{\text{IO}} \subset Q$. If $H \in \mathscr{H}_{\text{IO}}$ then there is a standard form IO hypergrammar $\mathbf{G} = ((N, T, F, \tau, P, S), \Rightarrow^{\text{IO}})$ generating H. Construct a quoted grammar structure $G' = (N, T, F', \tau', P', S)$ where

$$F' = F \cup \{p, q\} \qquad \text{where } p, q \text{ are new function symbols} \notin F \cup T,$$
$$\tau'(f) = \tau(f) \qquad \text{if} \quad f \in F,$$
$$\tau'(p) = 0 \qquad \text{and} \qquad \tau'(q) = 2,$$

and P' is constructed from P as follows.

If $\rho \in P$ involves neither union nor $\varnothing$-symbols on the right-hand side, then $\rho \in P'$.

If $\rho \in P$ is of the form $f(X_1,..., X_{\tau(f)}) \to g(X_1,..., X_{\tau(f)}) \cup h(X_1,..., X_{\tau(f)})$, then P' contains the productions

$$f(X_1,..., X_{\tau(f)}) \to q(g(X_1,..., X_{\tau(f)}), h(X_1,..., X_{\tau(f)})),$$
$$q(X_1, X_2) \to \langle q(X_1, X_2)\rangle,$$
$$q(X_1, X_2) \to X_1, q(X_1, X_2) \to X_2.$$

If $\rho \in P$ is of the form $f(X_1,..., X_{\tau(f)}) \to \varnothing$, then $f(X_1,..., X_{\tau(f)}) \to \langle p\rangle$ is in P'.

Then it can be shown that the quoted language generated by $\mathbf{G}'$, $Q(\mathbf{G}')$, is equal to the IO hyperlanguage $H((G, \Rightarrow^{IO}))$. The detailed proof is given in [7].

(2) $Q \subset \mathscr{H}_{IO}$. If $L \in Q$ then there is a normal form quoted grammar structure $G = (N, T, F, \tau, P, S)$ generating L. The productions of G are of one of the following three forms:

(1) $f(X_1,..., X_{\tau(f)}) \to g(h_1(X_1,..., X_{\tau(f)}),..., h_{\tau(g)}(X_1,..., X_{\tau(f)}))$,

(2) $f(X_1,..., X_{\tau(f)}) \to \eta$, $\eta \in (N \cup T)^*$,

(3) $f(X_1,..., X_{\tau(f)}) \to \langle g(X_1,..., X_{\tau(f)})\rangle$.

Construct an IO hypergrammar with HGS $G' = (N, T, F', \tau', P', S)$ from G as follows:

$$F' = F \cup \{f_q \mid f \in F\} \text{ where } \{f_q \mid f \in F\} \cap F = \varnothing,$$
$$\tau'(f) = \tau(f) \quad \text{and} \quad \tau'(f_q) = \tau(f), \quad \forall f \in F.$$

If $\rho \in P$ is a production of type (1), (2) above then $\rho \in P'$. If $\rho \in P$ is a production of type (3), say

$$f(X_1,..., X_{\tau(f)}) \to \langle g(X_1,..., X_{\tau(f)})\rangle,$$

then P' contains the productions

$$f(X_1,..., X_{\tau(f)}) \to g_q(X_1,..., X_{\tau(f)}),$$
$$g_q(X_1,..., X_{\tau(f)}) \to g_q(X_1,..., X_{\tau(f)}) \cup g_q(X_1,..., X_{\tau(f)}),$$
$$g_q(X_1,..., X_{\tau(f)}) \to g(X_1,..., X_{\tau(f)}),$$
$$g_q(X_1,..., X_{\tau(f)}) \to \varnothing.$$

It can be shown that the quoted language generated by $\mathbf{G}$, $Q(\mathbf{G})$, is equal to the IO hyperlanguage, $H((G', \Rightarrow^{IO}))$. Again, details are found in [7].

If every function symbol in a quoted grammar is quoted then the language generated is an OI macrolanguage. Bearing in mind the association between "quoting" and "unioning," it seems natural to ask if the class of OI macrolanguages is the subset of the languages defined by the IO hypergrammars where, for each function symbol f, there is only one production with f on the left-hand side. In particular, if the productions in some IO hypergrammar with f on the left-hand side are $f(X_1,..., X_{\tau(f)}) \to \gamma_1,..., f(X_1,..., X_{\tau(f)}) \to \gamma_n$, it is interesting to consider what happens if these productions are replaced by a single production $f(X_1,..., X_{\tau(f)}) \to \gamma_1 \cup \cdots \cup \gamma_n$. Of course one thing

that could happen is that some γ_i might contain the function symbol f, i.e., the production is self-recursive. If this were the case, then any hyperterm involving f could never generate any hyperterm which itself did not involve f. To avoid this situation, the production $f(X_1,\ldots, X_{\tau(f)}) \to \varnothing$ is also allowed.

DEFINITION 3.10. If for every function symbol f in a hypergrammar there is only one production with $f(X_1,\ldots, X_{\tau(f)})$ on the left-hand side and a term other than $\varnothing$ on the right-hand side, then the hypergrammar is said to have *unique productions*.

The following theorem is then a not unexpected result.

THEOREM 3.4. *$H \in \mathscr{H}_{\mathrm{OI}} = \mu_{\mathrm{OI}}$ iff there exists an* OI *hypergrammar with unique productions generating H iff there exists an* IO *hypergrammar with unique productions generating H.*

Proof. [7].

4. CLOSURE PROPERTIES

THEOREM 4.1. *$\mathscr{H}_{\mathrm{unr}}$, $\mathscr{H}_{\mathrm{OI}}$, and $\mathscr{H}_{\mathrm{IO}}$ are all closed under union, concatenation, Kleene closure, arbitrary homomorphisms, reversal, and substitution.*

Proof. The proof of this theorem is based on simple constructions which are independent of the mode of derivation.

Let $G = (N, T, F, \tau, P, S)$ and $G' = (N', T', F', \tau', P', S')$ be two hypergrammar structures which under the same mode of derivation, $\Rightarrow_G$, generate hyperlanguages H and H'. It is assumed, without loss of generality, that $F \cap F' = \varnothing$, $F' \cap T = \varnothing$, and $F \cap T' = \varnothing$.

Let $\tilde{S}$ be a new symbol not in $F \cup F' \cup T \cup T'$ and define $\tilde{\tau}$ by

$$\tilde{\tau}(\tilde{S}) = 0,$$

$$\tilde{\tau}(f) = \begin{cases} \tau(f), & \text{if } f \in F, \\ \tau'(f), & \text{otherwise.} \end{cases}$$

The required constructions are

Union. $H \cup H'$ is generated by the HGS $\tilde{G} = (N \cup N',\ T \cup T',\ F \cup F' \cup \{\tilde{S}\}, \tilde{\tau}, \tilde{P}, \tilde{S})$ under $\Rightarrow_G$ where $\tilde{P} = P \cup P' \cup \{\tilde{S} \to S \cup S'\}$.

Concatenation. HH' is generated by the HGS $\tilde{G} = (N \cup N',\ T \cup T', F \cup F' \cup \{\tilde{S}\}, \tilde{\tau}, \tilde{P}, \tilde{S})$ under $\Rightarrow_G$, where $\tilde{P} = P \cup P' \cup \{\tilde{S} \to SS'\}$.

Kleene closure. H^* is generated by the HGS $G^* = (N, T, F, \tau, P^*, S)$ under $\Rightarrow_G$, where $P^* = P \cup \{S \to SS\} \cup \{S \to \epsilon\}$.

Arbitrary homomorphism, α: $T \to W$. $\alpha(H)$ is generated by $(N, W, F, \tau, P^\alpha, S)$ under $\Rightarrow_G$, where P^α is constructed from P by replacing every occurrence of a terminal symbol, a, in a production of P by $\alpha(a)$.

Reversal. Let $\psi: \mathscr{F}_\Sigma(\hat{T} \cup N) \to \mathscr{F}_\Sigma(\hat{T} \cup N)$ be the reversal function defined by

(1) $\psi(\varnothing) = \varnothing$,

(2) $\psi(a) = a$, $\forall a \in N \cup T \cup \{\epsilon\}$,

(3) $\psi(t_1 t_2) = \psi(t_2)\,\psi(t_1)$ and $\psi(t_1 \cup t_2) = \psi(t_1) \cup \psi(t_2)$, $\forall t_1, t_2 \in \mathscr{F}_\Sigma(\hat{T} \cup N)$,

(4) $\psi(f(t_1,\ldots, t_n)) = f(\psi(t_1),\ldots, \psi(t_n))$, $\forall t_1, t_2,\ldots, t_n \in \mathscr{F}_\Sigma(\hat{T} \cup N)$.

$\psi(H)$ is then generated by the HGS $G^\psi = (N, T, F, \tau, P^\psi, S)$ under $\Rightarrow_G$, where P^ψ is constructed from P by replacing each production $f(X_1,\ldots, X_{\tau(f)}) \to \gamma$ by $f(X_1,\ldots, X_{\tau(f)}) \to \psi(\gamma)$. $\psi(H) = \{\psi(x) \mid x \in H\}$ is the reversal of H.

Substitution. Let α be any substitution and let $T = \{a_1,\ldots, a_n\}$. It can be assumed that $\alpha(a_i)$ is generated by the HGS $G_i = (N_i, T_i, F_i, \tau_i, P_i, S_i)$ under $\Rightarrow_G$, where $F_i \cap F_j = \varnothing$ and $F_i \cap T_j = \varnothing$ for $i \neq j$ and $F_i \cap F = \varnothing$, $F_i \cap \{a_1,\ldots, a_n\} = \varnothing$, $T_i \cap F = \varnothing$ for all $i = 1,\ldots, n$. If $\alpha(a_i) = \varnothing$, then assume that P_i consists of the single production $S_i \to \varnothing$.

Define a new HGS, $G^\alpha = (N^\alpha, T^\alpha, F^\alpha, \tau^\alpha, P^\alpha, S)$, where

$$N^\alpha = N \cup N_1 \cup \cdots \cup N_n,$$

$$T^\alpha = T_1 \cup \cdots \cup T_n,$$

$$F^\alpha = F \cup F_1 \cup \cdots \cup F_n,$$

$$\tau^\alpha(f) = \begin{cases} \tau(f), & \text{if } f \in F, \\ \tau_i(f), & \text{if } f \in F_i, \quad i = 1,\ldots, n. \end{cases}$$

Now, define a function $h: \mathscr{F}_\Sigma(\hat{T} \cup N) \to \mathscr{F}_{\Sigma'}(N \cup \{\epsilon, \varnothing\})$ where $\Sigma = F \cup \{\textit{concat}, \textit{union}\}$ and $\Sigma' = \Sigma \cup \{S_i \mid i = 1,\ldots, n\}$, by

(1) $h(a_i) = S_i$, $i = 1,\ldots, n$,

(2) $h(\varnothing) = \varnothing$, $h(\epsilon) = \epsilon$,

(3) $h(X) = X$, $\forall X \in N$,

(4) $h(t_1 t_2) = h(t_1)\,h(t_2)$ and $h(t_1 \cup t_2) = h(t_1) \cup h(t_2)$, $\forall t_1, t_2 \in \mathscr{F}_\Sigma(\hat{T} \cup N)$,

(5) $h(f(t_1,\ldots, t_{\tau(f)})) = f(h(t_1),\ldots, h(t_{\tau(f)}))$, $\forall f \in F$ and $\forall t_1,\ldots, t_{\tau(f)} \in \mathscr{F}_\Sigma(\hat{T} \cup N)$.

P^α is then defined to contain $P_1 \cup P_2 \cup \cdots \cup P_n$ and also productions constructed from P as follows:

$$\text{if } f(X_1,\ldots, X_n) \to \gamma \text{ is in } P \text{ then } f(X_1,\ldots, X_n) \to h(\gamma) \text{ is in } P^\alpha.$$

Also P^α contains the productions $S_i \to S_i \cup S_i$ for $i = 1, 2,\ldots, n$. These productions enable any S_i in P^α to generate subsets of $\alpha(a_i)$.

Then it is clear that the language generated by G^α under $\Rightarrow_G$ is precisely $\alpha(H)$. An interesting corollary is

THEOREM 4.2. $\mathscr{H}_{\text{unr}} \neq \mathscr{L}_1$; $\mathscr{H}_{\text{IO}} \neq \mathscr{L}_1$; $\mathscr{H}_{\text{OI}} \neq \mathscr{L}_1$.

Proof. $\mathscr{L}_1$, the class of context sensitive languages, is not closed under substitution but only under ϵ-free substitution.

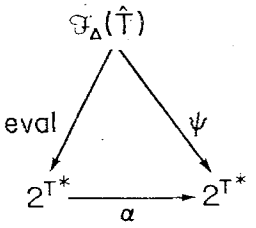
eval: $\mathscr{F}_\Delta(\hat{T}) \to 2^{T^*}$ has been defined as the Δ-algebra homomorphism defined by $\text{eval}(a) = \{a\}$. If $\mathscr{T}$ is the subset of the carrier of $\mathscr{F}_\Delta(\hat{T})$ generated by a hypergrammar then $\bigcup_{x\in\mathscr{T}} \text{eval}(x)$ is a hyperlanguage. It is interesting to try and generalize this definition.

DEFINITION 4.1. $L \subset T^*$ is a *generalized hyperlanguage* if there exists a Δ-algebra homomorphism ψ: $\mathscr{F}_\Delta(\hat{T}) \to 2^{T^*}$ and a subset $\mathscr{T}$ of the carrier of $\mathscr{F}_\Delta(\hat{T})$, which is exactly the set of all terminal terms generated by a hypergrammar, such that $L = \bigcup_{x\in\mathscr{T}} \psi(x)$.

If the Δ-algebra homomorphism α: $2^{T^*} \to 2^{T^*}$ is defined by

$$\alpha(\{a\}) = \psi(a), \qquad \forall a \in T \cup \{\epsilon\},$$
$$\alpha(\{ax\}) = \psi(a) \cdot \psi(x), \qquad \forall a \in T, \quad x \in T^*,$$
$$\alpha(A) = \bigcup_{x\in A} \alpha(\{x\}), \qquad \forall A \in 2^{T^*}.$$

Then the diagram



commutes. Thus,

$$\bigcup_{x\in\mathscr{T}} \psi(x) = \bigcup_{x\in\mathscr{T}} \alpha(\text{eval}(x))$$
$$= \alpha\left(\bigcup_{x\in\mathscr{T}} \text{eval}(x)\right).$$

So a generalized hyperlanguage is a homomorphic image of a hyperlanguage and so is a hyperlanguage.

5. The Evaluation of Sentential Forms

The evaluation of terminal terms generated by a hypergrammar has been defined using a map eval: $\mathscr{F}_\Delta(\hat{T}) \to 2^{T^*}$. This map can be extended in a natural way to produce an evaluation eval: $\mathscr{F}_\Sigma(\hat{T}) \to 2^{T^*}$ on all sentential forms of the hypergrammar.

2^{T^*} has been regarded as a Δ-algebra. $\Sigma = \Delta \cup F$, so if we define the action of the operators F on 2^{T^*}, 2^{T^*} can be regarded as a Σ-algebra. Define $f(A_1, \ldots, A_{\tau(f)}) = \varnothing$, $\forall A_1, \ldots, A_{\tau(f)} \in 2^{T^*}$ and $\forall f \in F$. eval is then the Σ-algebra homomorphism defined by $\text{eval}(\varnothing) = \varnothing$ and $\text{eval}(a) = \{a\}$, $\forall a \in T \cup \{\epsilon\}$.

DEFINITION 5.1. If $\mathbf{G} = (G, \Rightarrow_G)$ is a hypergrammar, then the *extended hyperlanguage* generated by $\mathbf{G}$, $E(\mathbf{G})$ is defined by

$$E(\mathbf{G}) = \bigcup \{\text{eval}(x) \mid S \Rightarrow_G^* x\}.$$

The extended language differs from the previous definition of language since all sentential forms are evaluated and not just terminal terms. Clearly $E(\mathbf{G}) \supset H(\mathbf{G})$.

The extended unrestricted hyperlanguages, extended IO hyperlanguages, and extended OI hyperlanguages are defined according to the mode of derivation of the hypergrammar. The classes of languages so defined are denoted, respectively, by $\mathscr{E}_{\text{unr}}$, $\mathscr{E}_{\text{IO}}$, $\mathscr{E}_{\text{OI}}$.

EXAMPLE 9. If a hypergrammar structure G has productions

$$S \to f(a, b, c),$$
$$f(X_1, X_2, X_3) \to f(aX_1, bX_2, cX_3) \cup X_1X_2X_3,$$

then $H((G, \Rightarrow^{\text{unr}})) = H((G, \Rightarrow^{\text{IO}})) = H((G, \Rightarrow^{\text{OI}})) = \varnothing$, and $E((G, \Rightarrow^{\text{unr}})) = E((G, \Rightarrow^{\text{IO}})) = E((G, \Rightarrow^{\text{OI}})) = \{a^nb^nc^n \mid n \geqslant 1\}$. Note, however, that if G' has productions

$$S \to f(a, b, c),$$
$$f(X_1, X_2, X_3) \to f(aX_1, bX_2, cX_3) \cup X_1X_2X_3,$$
$$f(X_1, X_2, X_3) \to \varnothing,$$

then $H((G', \Rightarrow^{\text{unr}})) = H((G', \Rightarrow^{\text{IO}})) = H((G', \Rightarrow^{\text{OI}})) = \{a^nb^nc^n \mid n \geqslant 1\}$, and $E((G', \Rightarrow^{\text{unr}})) = E((G', \Rightarrow^{\text{IO}})) = E((G', \Rightarrow^{\text{OI}})) = \{a^nb^nc^n \mid n \geqslant 1\}$.

This example shows that if $\mathbf{G}$ and $\mathbf{G}'$ are hypergrammars then $E(\mathbf{G}) = E(\mathbf{G}')$ does not necessarily imply that $H(\mathbf{G}) = H(\mathbf{G}')$. Also there are hypergrammars $\mathbf{G}$ and $\mathbf{G}'$ such that $H(\mathbf{G}) = H(\mathbf{G}')$ but $E(\mathbf{G}) \neq E(\mathbf{G}')$, for consider the HGS G with only one production, $S \to S \cup a$ and the HGS G' also with only one production, $S \to S \cup b$, then under any mode of derivation the extended hyperlanguages generated by G and G' are $\{a\}$ and $\{b\}$, respectively, although $H((G, \Rightarrow_G)) = H((G', \Rightarrow_G)) = \varnothing$.

THEOREM 5.1. $\mathscr{E}_{\text{unr}} = \mathscr{H}_{\text{unr}} = \mathscr{E}_{\text{OI}} = \mathscr{H}_{\text{OI}}$; $\mathscr{E}_{\text{IO}} = \mathscr{H}_{\text{IO}}$.

Proof. [7].

Thus for any class of hypergrammars, the class of languages defined is unchanged if the evaluation is defined on all sentential forms and not just on terminal terms.

6. THREE HIERARCHIES OF LANGUAGES

By placing restrictions upon the arities of the function symbols of hypergrammars three new hierarchies of languages are found, each embedded in the class of IO hyperlanguages. These hierarchies are hardly surprising but nevertheless are interesting because they can give a measure of the "complexity" of a given language.

DEFINITION 6.1. Let $G = (N, T, F, \tau, P, S)$ be a hypergrammar structure. Then (F, τ) is a stratified alphabet of function symbols. $M = \max\{\tau(f) \mid f \in F\}$ is defined to be the *arity* of G. $\mathscr{H}_{IO,n}$, $\mathscr{H}_{OI,n}$, $\mu_{IO,n}$, and $\mu_{OI,n}$ denote, respectively, the classes of languages generated by IO hypergrammars, OI hypergrammars, IO macrogrammars, and OI macrogrammars of arity n.

THEOREM 6.1. (1) $\bigcup_{n=0}^{\infty} \mathscr{H}_{IO,n} = \mathscr{H}_{IO}$; $\bigcup_{n=0}^{\infty} \mathscr{H}_{OI,n} = \mathscr{H}_{OI}$; $\bigcup_{n=0}^{\infty} \mu_{IO,n} = \mu_{IO}$; $\bigcup_{n=0}^{\infty} \mu_{OI,n} = \mu_{OI}$.

(2) $\mathscr{H}_{IO,0} = \mathscr{H}_{OI,0} = \mu_{IO,0} = \mu_{OI,0} = \mathscr{L}_2$, *the class of context-free languages.*

(3) *For all* $n \geqslant 0$, $\mathscr{H}_{IO,n}$, $\mathscr{H}_{OI,n}$, $\mu_{IO,n}$, $\mu_{OI,n}$ *are closed under union, concatenation, Kleene closure, arbitrary homomorphism, and reversal.*

(4) *For all* $n \geqslant 0$, $\mathscr{H}_{OI,n} = \mu_{OI,n}$.

(5) *For all* $n \geqslant 0$, $\mathscr{H}_{IO,n} \subsetneqq \mathscr{H}_{IO,n+1}$; $\mathscr{H}_{OI,n} \subsetneqq \mathscr{H}_{OI,n+1}$; $\mu_{IO,n} \subsetneqq \mu_{IO,n+1}$; $\mu_{OI,n} \subsetneqq \mu_{OI,n+1}$.

(6) *For all* $n \geqslant 1$, *there exists* $L \in \mu_{IO,n}$ *such that* $L \notin \mu_{OI,m}$ *for any* m *and there exists* $L' \in \mu_{OI,n}$ *such that* $L' \notin \mu_{IO,m}$ *for any* m.

(7) $\mu_{IO,n} \cup \mu_{OI,n} \subsetneqq \mathscr{H}_{IO,n}$.

Proof. (1) and (2) follow immediately from the definitions.

(3) and (4) require proofs following identical lines as those of Theorems 3.2 and 4.1.

(5) The proof of containment is trivial but the proof of the inequality is not so easy. It is shown in [7] that the language $L_r = \{a_1^i a_2^i \cdots a_r^i \mid i \geqslant 1\}$ is an element of any one of the classes of languages $\mathscr{H}_{IO,n}$, $\mathscr{H}_{OI,n}$, $\mu_{IO,n}$, $\mu_{OI,n}$ if and only if $r \leqslant 2n + 2$.

(6) The language L' of Example 3 $\in \mu_{OI,1}$ and hence $\in \mu_{OI,n}$ $\forall n \geqslant 1$ but $L' \notin \mu_{IO}$.

The language of Example 4 $\in \mu_{IO,1}$ and hence $\in \mu_{IO,n}$ $\forall n \geqslant 1$ but $\notin \mu_{OI}$.

(7) $\mu_{IO,n} \subset \mathscr{H}_{IO,n}$ and $\mu_{OI,n} \subset \mathscr{H}_{OI,n}$ follows since a macrogrammar is just a hypergrammar with no union or $\varnothing$ symbols. Let $L \in \mu_{IO,n}$ and $L' \in \mu_{OI,n}$ be the languages defined in (6) above. $L \subset \{1, c\}^*$ and $L' \subset \{a, b\}^*$. Define $\tilde{L} = L \cup L'$. Then by part (3),

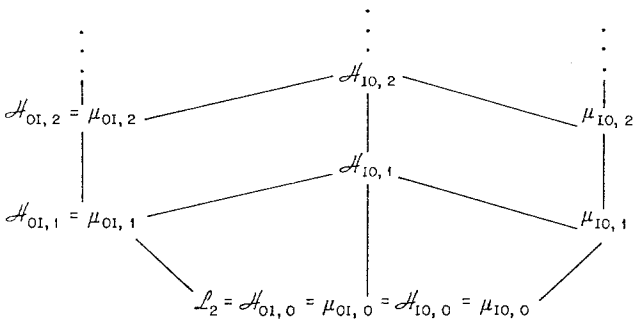


FIGURE 1

$\tilde{L} \in \mathscr{H}_{\mathrm{IO},n}$. $\tilde{L} \notin \mu_{\mathrm{IO},n}$ for $\{a, b\}^*$ is a regular set and if $\tilde{L} \in \mu_{\mathrm{IO},n}$, then since μ_{IO} is closed under intersection with regular sets, this would imply $\tilde{L} \cap \{a, b\}^* \in \mu_{\mathrm{IO}}$, i.e., $L' \in \mu_{\mathrm{IO}}$—a contradiction. Also, $\tilde{L} \notin \mu_{\mathrm{OI},n}$ for $\{1, c\}^*$ is a regular set and if $\tilde{L} \in \mu_{\mathrm{OI},n}$ then since μ_{OI} is closed under intersection with regular sets, this would imply $\tilde{L} \cap \{1, c\}^* \in \mu_{\mathrm{OI}}$, i.e., $L \in \mu_{\mathrm{OI}}$—again a contradiction.

The hierarchies of languages are described in Fig. 1. A line denotes proper containment, i.e., the class of languages at the upper end of the line strictly includes the class of languages at the lower end of the line.

7. A Lattice Theoretic Approach

The fact that any context-free language $L \subset T^*$ can be expressed as a fixed point of a set of equations over the lattice $\mathscr{L} = 2^{T^*}$ is well known. Blikle [2] has produced some interesting generalized results from this lattice theoretic approach. Dana Scott's work in the theory of computation [9] has resulted in the study of lattices of continuous functions from one lattice to another. If $\mathscr{C}_n$ is defined to be the lattice of continuous functions $(2^{T^*})^n \to 2^{T^*}$ then any OI hyperlanguage of order $n \subset T^*$ can be expressed as a fixed point of some set of equations over the lattice $\mathscr{C}_n$. Furthermore, any IO hyperlanguage of order $n \subset T^*$ can be expressed as a fixed point of a set of equations over the lattice $2^{\mathscr{C}_n}$. Many of the theorems obtained by Blikle [2] can now be used to produce interesting corollaries. A generalized normal form theorem is produced in [7] and overall the foundations of a theory of formal languages based on lattices appear to have been laid. Many questions still remain open; for instance, can other previously studied formal languages be expressed as fixed points of equation sets over lattices? What other theorems can be proved in terms of lattices? What type of languages are defined by fixed points of equation sets over other particular lattices?

References

1. A. V. Aho, Indexed grammars: An Extension of context-free grammars, *J. Assoc. Comput. Mach.* **15** (1968), 647–671.
2. A. Blikle, Equational language, *Inform Contr.* **21** (1972), 134–147.
3. N. Chomsky, Three models for the description of language, *PGIT* **2** (1956), 113–124.
4. N. Chomsky, On certain formal properties of grammars, *Inform. Contr.* **2** (1959), 137–167.
5. M. J. Fischer, Grammars with macro-like productions, Doctoral Dissertation, Harvard University, May, 1968. *Reprinted in* Mathematical linguistics and automatic translation, Harvard University Computation Laboratory Report, NSF-22.
6. S. Greibach and J. E. Hopcroft, Independence of AFL operations, *in* "Studies in Abstract Families of Languages," Memoirs Vol. 87, pp. 33–40, Amer. Math. Soc., Providence, R.I., 1969.
7. V. J. Rayward-Smith, From languages to equation sets on lattices, Doctoral Dissertation, University of London, November, 1973.
8. D. J. Rosenkrantz, Programmed grammars and classes of formal languages, *J. Assoc. Comput. Mach.* **16** (1969), 107–131.
9. D. Scott, Outline of a mathematical theory of computation, *in* "Proc. 4th Annual Princeton Conference on Information Sciences and Systems (1970)," pp. 169–176.