DEPT. OF COMPUTER SCIENCE,
UNIVERSITY OF WATERLOO,
WATERLOO, ONTARIO N2L 3G1

PARALLEL REWRITING SYSTEMS ON TERMS

Karel Culik II
and
T.S.E. Maibaum

CS-74-04

Department of Applied Analysis and
Computer Science
University of Waterloo
Waterloo, Ontario, Canada
March 1974

Faculty of Mathematics

University of Waterloo

Waterloo, Ontario

Canada

Department of Applied Analysis

&

Computer Science

PARALLEL REWRITING SYSTEMS ON TERMS

Karel Culik II
and
T.S.E. Maibaum

CS-74-04

Department of Applied Analysis and
Computer Science
University of Waterloo
Waterloo, Ontario, Canada
March 1974

The research for this report was supported by National Research Council grant numbers A5549 and A7403.

0.

A. Lindenmayer introduced in [3] a mathematical model for developmental systems in biology. The simplest of these models were the so-called OL systems discussed in [4]. The properties which distinguish OL systems (and their generalisations) from the more common grammars of formal language theory are:

(i) Each symbol in the alphabet appears on the left-hand side of some production in the system;

(ii) A step in a derivation is accomplished by applying in parallel a production to each symbol in a string.

OL systems were generalised in [10] to the so-called TOL systems in which there are a number of tables of productions and each step in a derivation uses productions from only one of these tables. In both OL and TOL systems, no auxiliary (non-terminal) symbols were allowed. These systems were generalised to EOL and ETOL systems in [11] by allowing non-terminal symbols. An equivalent definition of EOL systems, the so-called FMOL systems, appeared independently in [2].

At the same time, the more classic parts of formal language theory were being extended in [5,6,7] to define context-free and indexed sets of terms (expressions) and to show that the derivation trees of a set of context-free terms is always a recognizable (regular) set of terms over a so-called derived alphabet. This resulted in the definition of a hierarchy of string languages with regular, context-free, and indexed sets of strings as the first three steps.

The obvious questions to ask are:

(i) Can we generalise OL, TOL, EOL and ETOL systems to terms (expressions)?

(ii) Can we then apply methods similar to those in [5,6,7] to obtain hierarchies of string languages based on these biologically motivated systems?

(iii) What would be the relationship of these hierarchies to each other and to the more classic hierarchy described in [5,6,7]?

Thus we begin in section 1 by defining EOL systems (and OL systems as a special case) over string alphabets. We then introduce many-sorted alphabets (a generalisation of ranked alphabets) and many-sorted algebras (a generalisation of algebras) as a setting for our generalisation. Regular and context-free grammars on many-sorted alphabets are introduced and a theorem connecting them is stated. The classical hierarchy is then introduced.

In section 2 we define EOL systems (and OL systems as a special case) over many-sorted alphabets. We then state a series of results which describe our hierarchies. In section 3 we summarise our results and note possible extensions.

Note that we can define an automaton which accepts exactly the language generated by an EOL system over a many-sorted alphabet. Unfortunately, this automaton is not very illuminating for the purposes of this study and a discussion of it is thus omitted.

1.

An EOL system G is a 4-tuple $\langle \Sigma, N, P, Z \rangle$ where

(i) $V = \Sigma \cup N$ and $\Sigma \cap N = \phi$;

(ii) Σ is called the terminal alphabet;

(iii) N is called the non-terminal alphabet;

(iv) $P \subseteq V \times V^*$ is called the set of productions of G. P is a total relation in V;

(v) Z is the axiom.

We define the relation $\overset{}{\underset{G}{\Rightarrow}} \subseteq V^* \times V^*$, called direct derivation, as follows for $w, w' \in V^*$: If $w = x_0 \ldots x_{n-1}$ with $x_j \in V$ for $0 \leq j \leq n-1$ and there exists $x_j \rightarrow v_j \in P$ for $0 \leq j \leq n-1$ and $w' = v_0 \ldots v_{n-1}$, then $w \underset{G}{\Rightarrow} w'$. Let $\overset{*}{\underset{G}{\Rightarrow}}$ be the reflexive, transitive closure of $\underset{G}{\Rightarrow}$. (We will often omit the G from $\underset{G}{\Rightarrow}$ when the grammar intended is obvious). Define the language generated by the EOL system G to be the set $L(G) = \{w \in \Sigma^* \mid Z \overset{*}{\Rightarrow} w\}$. L(G) is said to be an EOL language over Σ. If $N = \phi$ we call G an OL grammar and L(G) an OL language. It is well known that the class of OL languages over some alphabet Σ is a proper subclass of the class of EOL languages over Σ. Also, the classes of OL and CF languages over Σ are incomparable (as classes). On the other hand, the class of CF languages over Σ is a proper subclass of the class of EOL languages over Σ.

Let I be any set, called the set of sorts. A many-sorted alphabet Σ sorted by I is an indexed family of sets indexed by $I^* \times I$. That is $\Sigma = \{\Sigma_{\langle w,i \rangle}\}_{\langle w,i \rangle \in I^* \times I}$. Σ is said to be finite if both I and the disjoint union of Σ are finite. (Note that string alphabets and the more common ranked

alphabets are special cases of many-sorted alphabets). $f \in \Sigma_{<w,i>}$ is said to be of type $<w,i>$ argument sort (arity) w, (target) sort i, and rank $\ell(w)$. A symbol of type $<\lambda,i>$ (λ is the empty string) is said to be a constant (or nullary) symbol of sort i.

Example 1

A Σ-algebra A_Σ (or just A if the alphabet is obvious from the context) is an indexed family of sets $A = \{A_i\}_{i\in I}$ together with an indexed family of assignments

$$\alpha_{<w,i>}:\Sigma_{<w,i>} \rightarrow (A^w \rightarrow A)$$

from symbols in $\Sigma_{<w,i>}$ to functions from $A^w = A_{w_0} \times \ldots \times A_{w_{n-1}}$ to A_i. $((A^w \rightarrow A_i)$ is the set of functions from A^w to $A_i)$. We commonly denote the image of $f \in \Sigma_{<w,i>}$ under $\alpha_{<w,i>}$ by f itself, unless the context is not obvious, in which case we use f_A. A is called the carrier of the algebra A_Σ.

Let A and B be Σ-algebras. A homomorphism $\psi:A \rightarrow B$ is an indexed set of functions $\{\psi_i:A_i \rightarrow B_i\}_{i\in I}$ which "preserve the structure" of the algebra A. That is, for any $f \in \Sigma_{<w,i>}$ and $(a_0,\ldots,a_{n-1}) \in A^w$ (i.e. $a_j \in A_{w_j}$ for $0 \le j \le n-1$), $\psi_i(f_A(a_0,\ldots,a_{n-1})) = f_B(\psi_{w_0}(a_0),\ldots,\psi_{w_{n-1}}(a_{n-1}))$. Monomorphisms, epimorphisms, isomorphisms, and endomorphisms are defined in the obvious way.

Let $X = \{X_i\}_{i\in I}$ be any indexed family of sets. The indexed family of sets of terms (or expressions or words) on the alphabet Σ and generators X, denoted by $W_\Sigma(X) = \{(W_\Sigma(X))_i\}_{i\in I}$, is the least family of sets satisfying:

(0) $X_i \cup \Sigma_{<\lambda,i>} \subseteq (W_\Sigma(X))_i$;

(i) For each $f \in \Sigma_{<w,i>}$ and $(t_0,\ldots,t_{n-1}) \in (W_\Sigma(X))^w$, $ft_0\ldots t_{n-1} \in (W_\Sigma(X))_i$.

If each $X_i = \phi$, we denote $W_\Sigma(\{\phi\}_{i\in I})$ by W_Σ. We can make W_Σ into a Σ-algebra (called the word algebra or algebra of expressions or totally free algebra) by the assignment of operations to $f \in \Sigma_{<w,i>}$ as follows:

$$f_{W_\Sigma(X)}(t_0,\ldots,t_{n-1}) = ft_0\ldots t_{n-1}.$$

Example 2

Let $w = w_0\ldots w_{n-1} \in I^*$. Consider the set $\{y_{0,w_0},\ldots,y_{n-1,w_{n-1}}\}$ where $y_{i,w_i} \notin I$ and $y_{i,w_i} \notin \Sigma$ for any $0 \le i \le n-1$.

Let $Y_w = \{y_{0,w_0},\ldots,y_{n-1,w_{n-1}}\}$ for some $w = w_0\ldots w_{n-1} \in I^*$. We say Y_w is indexed by w. We can sort Y_w by I in the following way $(Y_w)_i = \{y_{j,i} \in Y_w | j < n\}$. We shall denote $W_\Sigma(\{(Y_w)_i\}_{i\in I})$ by $W_\Sigma(Y_w)$.

Theorem 1 (Fundamental Theorem of Algebra)

Let A be any Σ-algebra, X any family of generators, and $\psi = \{\psi_i\}_{i\in I}$ any indexed family of assignments $\{\psi_i : X_i \to A_i\}$. Then ψ extends in a unique way to a homomorphism $\bar{\psi}: W_\Sigma(X) \to A$. In particular, there is a unique homomorphism from W_Σ to A. □

We now proceed to define derived algebras and derived alphabets. Suppose we are given Σ(sorted by I). Let $D(I) = \{<w,i> | w$ is the arity of some $f \in \Sigma$ and $i \in I\}$. That is, the set D(I) is just the subset

of $I^* \times I$ with the first argument an arity of a symbol in Σ. We use $D(I)$ to sort an alphabet $D(\Sigma)$, called the derived alphabet of Σ, which is defined in the following way:

(i) If $f \in \Sigma_{<w,i>}$ then $f \in (D(\Sigma))_{<\lambda,<w,i>>}$. That is, f is a nullary of type $<\lambda,<w,i>>$ in $D(\Sigma)$;

(ii) For each w an arity of some symbol in Σ, $\ell(w) = n > 0$, let $\delta_w^j \in (D(\Sigma))_{<\lambda,<w,w_{j-1}>>}$ for $w = w_0 \ldots w_{n-1}$ and $1 \leq j \leq n$. These symbols are called projection symbols;

(iii) For each $<w,v,i> \in I^+ \times I^* \times I$, let $c_{<w,v,i>} \in (D(\Sigma))_{<<w,i><v,w_0>\ldots<v,w_{n-1}>,<v,i>>}$.

These are called composition symbols. We define an algebra $D(W_\Sigma)$, called the derived algebra of Σ, as follows:

(i) The carrier of $D(W_\Sigma)$ of sort $<w,i> \in D(I)$ is the set $(W_\Sigma(X_w))_i$. That is, $(D(W_\Sigma))_{<w,i>} = (W_\Sigma(X_w))_i$;

(ii) The assignment of operations to $D(\Sigma)$ is done as follows:

(a) Assign to $c_{<w,v,i>}$ an operation of composition with first argument of sort $<w,i>$, n arguments of sort $<v,w_j>$ for $0 \leq j \leq n-1$ and result of sort $<v,i>$;

(b) Assign to $f \in (D(\Sigma))_{<\lambda,<w,i>>}$, where $f \in \Sigma_{<w,i>}$, the constant $fx_{0,w_0} \ldots x_{n-1,w_{n-1}}$;

(c) Assign to $\delta_w^j \in (D(\Sigma))_{<\lambda,<w,w_{j-1}>>}$ the operation of projection. That is, given $c_{<w,v,w_{j-1}>}$ and $t_k \in (D(W_\Sigma))_{<v,w_k>}$ for $0 \leq k \leq n-1$, then $c_{<w,v,i>}(\delta_w^j, t_0, \ldots, t_{n-1}) = t_{j-1}$.

Example 3

Denote the unique homomorphism from $W_{D(\Sigma)}$ to $D(W_\Sigma)$ by $\mathrm{YIELD}: W_{D(\Sigma)} \to D(W_\Sigma)$.

A context free grammar G over a many-sorted alphabet Σ is a 4-tuple $\langle \Sigma, N, P, Z \rangle$ such that:

(i) $V = \Sigma \cup N$ and $\Sigma \cap N = \phi$;

(ii) Σ is called the terminal alphabet;

(iii) N is called the non-terminal alphabet;

(iv) P is a set of productions of the form $A(x_0, w_0, \ldots, x_{n-1}, w_{n-1}) \to t$ where $A \in N_{\langle w,i \rangle}$ and $t \in (W_V(X_w))_i$;

(v) Z is the axiom.

Now we define the relation of direct derivation for a CFG G. Let $\mathrm{Sub}_w(\_; t_0, \ldots, t_{n-1}): W_V(X_w) \to W_V$ be the (unique) homomorphism generated by the assignments $\psi_{w_j}: x_j, w_j \to t_j$ for $0 \le j \le n-1$. Intuitively, $s \overset{=}{\underset{G}{\Rightarrow}} s'$ if s has a subterm of the form $At_0 \ldots t_{n-1}$ and s' has a subterm of the form $\mathrm{Sub}_w(t; t_0, \ldots, t_{n-1})$ in its place.

Formally, $\overset{=}{\underset{G}{\Rightarrow}} \subseteq W_V \times W_V$ as follows, for $s, s' \in (W_\Sigma)_j$, some $j \in I$: $s \overset{=}{\underset{G}{\Rightarrow}} s'$ if and only if there exists a production in P, $\bar{s} \in (W_V(X_i))_j$ (with i being considered as the string of length one consisting of the symbol i), and $(t_0, \ldots, t_{n-1}) \in (W_V)^w$ such that

(i) $\mathrm{Sub}_i(\bar{s}; \mathrm{Sub}_w(A x_{0,w_0} \ldots x_{n-1,w_{n-1}}; t_0, \ldots, t_{n-1})) = s$

and (ii) $\mathrm{Sub}_i(\bar{s}; \mathrm{Sub}_w(t; t_0, \ldots, t_{n-1})) = s'$.

Let $\overset{*}{\underset{G}{\Rightarrow}}$ be the reflexive, transitive closure of $\underset{G}{\Rightarrow}$. (We will often omit the G if it is obvious from the context). The language generated by a context-free grammar $G = \langle \Sigma, V, P, Z \rangle$ is the indexed family of sets $L(G) = \{\{t \in (W_\Sigma)_i \mid Z \overset{*}{\Rightarrow} t\}_i\}_{i \in I}$. Such an L(G) is said to be context-free.

A context-free grammar G is said to be <u>regular</u> if $N_{\langle w,i \rangle} = \phi$ for $w \neq \lambda$. That is, only constant (or nullary) non-terminals are allowed. The set generated by a regular grammar is said to be a regular language.

It is well known that the class of regular languages over Σ is a proper subclass of the class of context-free sets over Σ.

<u>Example 4</u>

The following fundamental theorem is proved in [6,7]:

<u>Theorem 2</u>

Let G be a context-free grammar over Σ. We can effectively find a regular grammar G' over $D(\Sigma)$ such that $YIELD_{\langle \lambda,i \rangle}(L(G')) = L(G)$ (assuming $L(G) \subseteq (W_\Sigma)_i$). Conversely, suppose G is a regular grammar over $D(\Sigma)$ and $L(G) \subseteq (W_{D(\Sigma)})_{\langle \lambda,i \rangle}$, some $i \in I$, then we can effectively find a context-free grammar G' over Σ such that $YIELD_{\langle \lambda,i \rangle}(L(G)) = L(G')$. □

<u>Example 5</u>

We will usually omit the subscript $\langle \lambda,i \rangle$ from $YIELD_{\langle \lambda,i \rangle}$ for convenience of notation. Let Σ be a string alphabet and let $D^n(\Sigma)$ (<u>the n-th derived alphabet of Σ</u>) be defined recursively by $D^0(\Sigma) = \Sigma$ and $D^{n+1}(\Sigma) = D(D^n(\Sigma))$. Let REG^n be the class of recognizable sets over

$D^n(\Sigma)$ and let $YIELD^n: W_{D^n(\Sigma)} \to D^n(W_\Sigma)$ be the unique homomorphism from the word algebra over $D^n(\Sigma)$ to the n-th derived algebra of W_Σ (defined recursively by $D^0(W_\Sigma) = W_\Sigma$ and $D^{n+1}(W_\Sigma) = D(D^n(W_\Sigma))$). We define the operator Y as a map from any class of languages over $D^n(\Sigma)$ (for any $n \geq 0$) to the class of languages over Σ as follows: Given U, a class of languages over $D^n(\Sigma)$, $Y(U) = \{YIELD^n(L) | L \in U\}$. Thus Y(U) is always a class of string languages.

In [6] and [9] it is shown that the language $\{a^{2^{\cdot^{\cdot^{\cdot^{2^k}}}}} \text{ n-times} \mid k \geq 0\}$ is in $Y(REG^{n+1})$ but not in $Y(REG^n)$ for $n \geq 1$. Using this fact and Theorem 2, we can prove the following important result:

Theorem 3 (Hierarchy)

$Y(REG^n)$ is a proper subclass of $Y(REG^{n+1})$ for $n \geq 1$.

Corollary Let CF^n be the class of context-free sets over $D^n(\Sigma)$. Then $Y(CF^n)$ is a proper subclass of $Y(CF^{n+1})$ for $n \geq 1$. □

2.

Let Σ be a many-sorted alphabet. We define an EOL system G to be a 4-tuple $\langle \Sigma, N, P, Z \rangle$ where:

(i) $V = N \cup \Sigma$, $N \cap \Sigma = \phi$;

(ii) Σ is the terminal alphabet;

(iii) N is the non-terminal alphabet;

(iv) P is a set of productions of the form $A(x_{0,w_0}, \ldots, x_{n-1,w_{n-1}}) \to t$ where $A \in N_{\langle w,i \rangle}$ and $t \in W_V(X_w)_i$. For each $\langle w,i \rangle \in I^* \times I$ and each $A \in V_{\langle w,i \rangle}$, there is (are) some production(s) in P with left-hand side $A(x_{0,w_0}, \ldots, x_{n-1,w_{n-1}})$;

(v) Z is the axiom.

Note that we defined an EOL system so that for every element of the alphabet V, there is a production in P with that element on the left-hand side. This property is called completeness and along with the difference in the definition of derivation, it distinguishes EOL systems from context-free grammars.

We define the relation of direct derivation $\overset{=}{\underset{G}{\Rightarrow}} \subseteq W_V \times W_V$ as follows for $t, t' \in (W_\Sigma)_i$ (some $i \in I$): $t \overset{=}{\underset{G}{\Rightarrow}} t'$ if and only if $t' = \bar{\varphi}_{<\lambda,i>}(t)$ where $\bar{\varphi}$ is the endomorphism on $D(W_\Sigma)$ generated by the following assignments: For each symbol in t, say $f \in V_{<w,j>}$ there is some $f(x_{0,w_0},\ldots,x_{n-1,w_{n-1}}) \to s$ in P for $s \in (W_\Sigma(X_w))_j$. Let $\varphi_{<w,j>}(f) = s$. The assignments to symbols not appearing in t can be chosen arbitrarily.

Intuitively, we are replacing each symbol in an expression t by the corresponding right-hand side of a production in P. Moreover, as in the string case, we are performing these replacements in parallel. That is, all symbols in t are replaced at the same time.

Because the endomorphism is generated by certain assignments, it is evaluated in a constructive way. If we consider a tree corresponding to some term, its image under $\bar{\varphi}$ is found by first finding the image of the leaves (from the assignments), then finding the images of the direct ancestors of the leaves and attaching the images of the leaves to them and so on. As a result, we can simulate the derivation by the following "bottom-up" algorithm:

Consider a tree corresponding to the expression t, which we will again call t. Mark each symbol with its level in the tree starting with 0 for the root, 1 for the direct descendent of the root, etc. Thus the highest label which can appear at some node of the tree, say k, will be exactly the depth of the tree. For m = k, k-1,...,0 perform the following operation, starting at $t^k = t$:

(o) For each node of t^m at depth m, say $f \in V_{<w,j>}$, with subtrees $t_\ell \in (W_V)_{w_\ell}$ for $0 \leq \ell \leq n-1$, replace $Sub_w(f(x_{0,w_0},\ldots,x_{n-1,w_{n-1}});t_0,\ldots,t_{n-1})$ in t by $Sub_w(s;t_0,\ldots,t_{n-1})$ where $f(x_{0,w_0},\ldots,x_{n-1,w_{n-1}}) \to s$ is in P. Moreover, all nodes at depth m will be replaced in parallel. Then $t^0 = t'$.

Example 6

Let $\overset{*}{\underset{G}{\Rightarrow}}$ be the reflexive transitive closure of $\underset{G}{\Rightarrow}$. (We will again omit the G when the system we mean is obvious). The EOL language generated by the system $G = <\Sigma,N,P,Z>$ is the indexed family of sets $L(G) = \{\{t \in (W_\Sigma)_i \mid Z \overset{*}{\Rightarrow} t\}_i\}_{i\in I}$. If $N = \phi$, we call G an OL system and L(G) an OL language. Note that, although we have defined EOL systems requiring the property of completeness, this was not in fact necessary. We used this definition so that OL systems would be a special case of EOL systems. It can easily be shown that if we do not require completeness for OL systems, we get a class of OL languages which is not the same as that defined above.

Theorem 4

Let Σ be a many-sorted alphabet. The class of OL languages over Σ is incomparable to the class of CF languages over Σ.

Proof Consider the one-sorted alphabet Σ with one binary symbol x and two nullary symbols a, b. Let $L = \{a, xab, xba\}$. Then any OL system $G = \langle \Sigma, \phi, P, Z \rangle$ will have to have one of a, xab, or xba as axiom. Suppose the axiom is a. Then $a \rightarrow xab$ (or, symmetrically $a \rightarrow xba$) must be in P. But then a, xab, xxabb, xxxabbb, etc. are all in L, a contradiction to our definition of L. Suppose xab is the axiom. Then the only way to get a in L is to have $xzy \rightarrow z$ (applied to the axiom to give a) or a similar production in P. In any case, applying this production to xba (which we must be able to do because of the completeness condition) will give us $b \in L$. This again is a contradiction. Now L is obviously a context-free set (since it is finite) but it is not an OL language.

On the other hand, we know from [7] that the YIELD of the OL language of Example 6 is $L' = \{a^{\hat{2}^{2^n}} \mid n \geq 0\}$. We also know from [7] that L' is not the YIELD of any CF language over any many-sorted alphabet (although $\{a^{2^n} \mid n \geq 0\}$ is). Thus the classes of CF and OL languages are incomparable. □

Theorem 5

The classes of OL languages and CF languages over Σ are both proper subclasses of the class of EOL languages over Σ.

Proof The fact that every OL language is an EOL language follows trivially from the definition of OL systems.

Let $G = \langle \Sigma, N, P, Z \rangle$ be a CF grammar. Consider the EOL system $G' = \langle \Sigma, N, P', Z \rangle$ where $P' = P \cup \{a(x_{0,w_0}, \ldots, x_{n-1,w_{n-1}}) \rightarrow a(x_{0,w_0}, \ldots, x_{n-1,w_{n-1}}) \mid a \in V_{\langle w,i \rangle}$ for all $\langle w,i \rangle \in I^* \times I\}$. It can easily be shown that $L(G') = L(G)$. Thus every CF language is an EOL language.

The fact that the inclusions are proper follows from the previous theorem. □

Consider some string alphabet Σ and $D^n(\Sigma)$ (the n-th derived alphabet of Σ) for $n > 0$. Let EOL^n and OL^n be the class of EOL and OL languages, respectively, over the alphabet $D^n(\Sigma)$. Using the definition of the operator Y given after Theorem 2, we prove:

Theorem 6

The classes $Y(CF^n)$ and $Y(OL^n)$ are incomparable.

Proof Again, we have that there are some finite sets in $Y(CF^n)$ but not in $Y(OL^n)$ for any n. On the other hand, we know from [7] and [12] that $L = a^{2^{\cdot^{\cdot^{\cdot^{2^k}}}}}$ n times$\mid k \geq 0\}$ is in $Y(CF^n)$ but $L' = \{a^{2^{\cdot^{\cdot^{\cdot^{2^k}}}}}$ n+1 times$\mid k \geq 0\}$ is not. But L' is in $Y(OL^n)$ because we can construct the appropriate system over $D^n(\Sigma)$ using a technique similar to that used in Example 6. This then proves our theorem.

Corollary (o) $Y(OL^n)$ is a proper subclass of $Y(EOL^n)$;

(i) $Y(CF^n)$ is a proper subclass of $Y(EOL^n)$.

Proof Follows trivially from definitions and the above theorem. □

Before we prove our next (and final) result, we will introduce the concept of indexed grammar on trees [8]. This is a simple generalisation of indexed grammars on strings [1]. The definition was motivated by [9] where indexed grammars on strings were generalised to generate any language in $Y(REG^n)$ for any $n > 0$.

An <u>indexed grammar</u> G over a many-sorted alphabet Σ is a 5-tuple $\langle \Sigma, N, F, P, Z \rangle$ such that:

(i) $N \cap \Sigma = \phi$, $V = N \cup \Sigma$;

(ii) Σ is the <u>terminal alphabet</u>;

(iii) N is the <u>non-terminal alphabet</u>;

(iv) F is a finite set each element of which is a finite set of ordered pairs of the form $A(x_{0,w_0},\ldots,x_{n-1,w_{n-1}}) \rightarrow t$ for $A \in N_{\langle w,i \rangle}$ and $t \in (W_V(X_w))_i$. An element $f \in F$ is called an <u>index</u> or <u>flag</u>;

(v) P is a set of productions of the form $A(x_{0,w_0},\ldots,x_{n-1,w_{n-1}}) \rightarrow t$ where $A \in N_{\langle w,i \rangle}$ and $t \in (W_{V;F}(X_w))_i$;

(vi) Z is the axiom.

$W_{V;F}(X_w)$ is the word algebra over the alphabet V;F which is defined as follows:

(i) $\Sigma \subseteq V;F$ and

(ii) For each $\langle w,i \rangle$ and $A \in N_{\langle w,i \rangle}$, $(F^*,A) \subseteq (V;F)_{\langle w,i \rangle}$.

(We will write (λ,A) as A). So $W_{V;F}(X_w)$ is like $W_V(X_w)$ except that nodes labelled by non-terminals are replaced by pairs which are made up of a string of flags and a non-terminal. Elements of V;F have the type of the corresponding symbol in V.

Direct derivation $\underset{G}{\Rightarrow} \subseteq W_{V;F} \times W_{V;F}$ is defined as follows:

(i) If $t = Sub_i[s;Sub_w[(y,A)(x_{0,w_0},\ldots,x_{n-1,w_{n-1}});t_0,\ldots,t_{n-1}]]$ for $A \in N_{<w,i>}$, $y \in F^*$, $s \in (W_{V;F}(X_i))_j$, $t_k \in (W_{V;F})_{w_k}$ $(0 \leq k \leq n-1)$, $t \in (W_{V;F})_j$ and $A(x_{0,w_0},\ldots,x_{n-1,w_{n-1}}) \rightarrow u \in P$ then

$$t \underset{G}{\Rightarrow} t'$$

if and only if $t' = Sub_i[s;Sub_w[u';t_0,\ldots,t_{n-1}]]$ where u' is obtained from u by left-concatenating y onto each first element of the pairs $V;F\text{-}\Sigma$ which appear in u;

(ii) If $t = Sub_i[s;Sub_w[(yf,A)\ (x_{0,w_0},\ldots,x_{n-1,w_{n-1}});t_0,\ldots,t_{n-1}]]$ (symbols have meaning as above with $f \in F$) and $A(x_{0,w_0},\ldots,x_{n-1,w_{n-1}}) \rightarrow u$ is in f then

$$t \underset{G}{\Rightarrow} t'$$

if and only if $t' = Sub_i[s;Sub_w[(y,u');t_0,\ldots,t_{n-1}]]$ where $u' \in (W_{V;F}(X_w))_i$ is obtained from $(W_V(X_w))_i$ by replacing each non-terminal B of u by (y,B).

$\overset{*}{\underset{G}{\Rightarrow}}$ is the reflexive, transitive closure of $\underset{G}{\Rightarrow}$. The language generated by an indexed grammar $G = <\Sigma,N,F,P,Z>$ is the indexed family of sets $L(G) = \{\{t \in (W_\Sigma)_i \mid Z \overset{*}{\underset{G}{\Rightarrow}} t\}_i\}_{i \in I}$. Such an $L(G)$ is said to be indexed.

Example 7

Lemma 1 (See [8]): Let G be an indexed grammar over Σ. We can effectively find a context free grammar G' over $D(\Sigma)$ such that $YIELD(L(G')) = L(G)$.

Conversely, suppose G is a context free grammar over $D(\Sigma)$ and $L(G) \subseteq \{(W_{D(\Sigma)})_{<\lambda,i>}\}_{i\in I}$, then we can effectively find an indexed grammar G' over Σ such that $\{YIELD_{<\lambda,i>}(L(G))\}_{i\in I} = L(G')$. □

Let IX^n be the class of indexed languages over $D^n(\Sigma)$. We then have the following corollaries to Theorem 3:

Corollary $Y(IX^n)$ is a proper subclass of $Y(IX^{n+1})$ for $n \geq 1$.

Corollary $Y(IX^n) = Y(CF^{n+1}) = Y(REG^{n+2})$ for $n \geq 1$. □

We are now ready to prove our final result:

Theorem 7 $Y(EOL^n)$ is a proper subclass of $Y(CF^{n+1})$ for all $n \geq 1$. (The fact that EOL is a proper subclass of $Y(CF^1)$ is proved in [11]).

Proof By the above corollaries, to prove inclusion it is sufficient to show that, given any EOL^n grammar $G = <D^n(\Sigma),N,P,Z>$, we can effectively find an IX^n grammar $G' = <D^n(\Sigma),N',F,P',Z'>$ such that $L(G) = L(G')$. So, given G, define G' as follows:

(i) $N' = N \cup \{Z'\}$ where Z' is of the same type(s) as Z;

(ii) $F = \{f,g\}$ where f is those productions in P which have some non-terminal on the right (the so-called non-terminal productions) and g is those productions in P which have no non-terminal on the right (the so-called terminal productions);

(iii) $P' = \{Z' \rightarrow (g,Z), Z \rightarrow (f,Z)\}$.

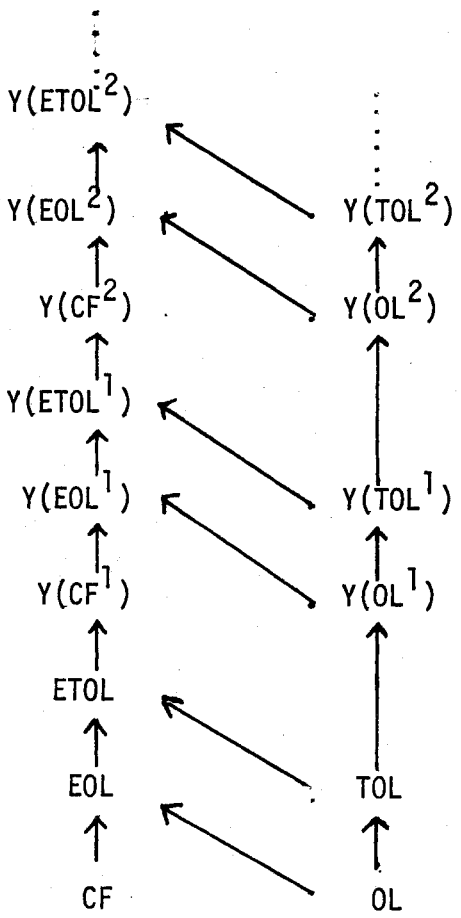
Thus we get $Z' \overset{*}{\underset{G'}{\Rightarrow}} (gf^n,Z) \overset{*}{\underset{G'}{\Rightarrow}} t$ as a derivation in G'. The f^n is used to simulate an n-step parallel derivation using only non-terminal productions (since the f^n goes in front of each non-terminal appearing in any expression

derived from Z). g is then used to simulate the final (parallel) switch to terminal symbols. (Note that we are using the version of EOL grammars which do not require productions involving terminal symbols on the left of productions). It should be clear that $Z' \overset{*}{\underset{G'}{\Rightarrow}} t$ if and only if $Z \overset{*}{\underset{G}{\Rightarrow}} t$ and that $L(G) = L(G')$.

For the proof of proper inclusion, the reader is referred to [2]. □

3.

The extensions of the above theory to TOL and ETOL systems over many-sorted alphabets is straightforward. We can integrate these extensions with the above results in the following diagram (where TOL^n and $ETOL^n$ have the obvious definitions):



In the diagram, a class L is a proper subclass of a class L' if and only if there is a directed path from L to L'. Otherwise, L and L' are incomparable. So, for instance, $Y(OL^n)$ and $Y(CF^n)$ are incomparable but are both proper subclasses of $Y(EOL^n)$, for any $n \geq 0$. Similarly, $Y(TOL^n)$ and $Y(EOL^n)$ are incomparable but both proper subclasses of $Y(ETOL^n)$, for any $n \geq 0$.

Note that we need not have started with a string alphabet Σ. We could start with any (finite) many-sorted alphabet Σ. In this case, our diagram would represent a hierarchy of term languages.

We conjecture that EOL^n are closed under all the AFL operations except inverse homomorphism.

Example 1

Let $I = \{0,1\}$, $\Sigma_{<\lambda,0>} = \{\lambda\}$, $\Sigma_{<\lambda,1>} = \{a\}$, $\Sigma_{<10,0>} = \{*\}$, $\Sigma_{<11,1>} = \{+\}$. Then Σ is a many-sorted alphabet. λ and a are nullaries of sorts 0 and 1, respectively. * is of type <10,0>, arity (argument sort) 10, (target) sort 0, and rank $\ell(10) = 2$. + is of type <11,1>, arity 11, sort 1 and rank 2. □

Example 2

Given the alphabet of Example 1 and the family of generators $X_0 = \phi$ and $X_1 = \{x\}$, we have $(W_\Sigma(X))_0 = \{\lambda,*a\lambda,*x\lambda,*+aa\lambda,*+ax\lambda,*+xa\lambda,*+xx\lambda,*+a+aa\lambda,\text{etc.}\}$ and $(W_\Sigma(X))_1 = \{a,x,+aa,+ax,+xa,+xx,+a+aa,\text{etc.}\}$. □

Example 3

Consider the alphabet Σ of Example 1. The derived algebra of W_Σ, $D(W_\Sigma)$, is obtained as follows:

(i) The sorting set is $D(I) = \{<\lambda,0>,<\lambda,1>,<10,0>,<10,1>,<11,0>,<11,1>\}$;

(ii) The indexed set of operator symbols is

$$D^{(\Sigma)}_{<\lambda,<\lambda,0>>} = \{\lambda\}, D^{(\Sigma)}_{<\lambda,<\lambda,1>>} = \{a\},$$

$$D^{(\Sigma)}_{<\lambda,<10,0>>} = \{\delta^2_{10},*\}, D^{(\Sigma)}_{<\lambda,<10,1>>} = \{\delta^1_{10}\},$$

$$D^{(\Sigma)}_{<\lambda,<11,0>>} = \phi,\ D^{(\Sigma)}_{<\lambda,<11,1>>} = \{\delta^1_{11},\delta^2_{11},+\} \text{ and}$$

$$c \in D^{(\Sigma)}_{<<w,i><v,w_0>\ldots<v,w_{n-1}>,<v,i>>} \quad \text{for each } (w,v,i) \in \{10,11\} \times \{\lambda,10,11\} \times I \text{ (with all other } D^{(\Sigma)}_{<w,i>} = \phi;$$

(iii) Let $X_0 = \{y_{1,0}\}$ and $X_1 = \{x_{0,1}, x_{1,1}, y_{0,1}\}$ be a family of generators indexed by I. Then the element of the carrier of $D(W_\Sigma)$ of sort $\langle\lambda,0\rangle$ is the set $(W_\Sigma)_0$, the element of sort $\langle\lambda,1\rangle$ is $(W_\Sigma)_1$, the element of sort $\langle 10,0\rangle$ is $W_\Sigma(\{y_{0,1}, y_{1,0}\})_0$, the element of sort $\langle 10,1\rangle$ is $W_\Sigma(\{y_{0,1}, y_{1,0}\})_1$, the element of sort $\langle 11,0\rangle$ is $W_\Sigma(\{x_{0,1}, x_{1,1}\})_0$ and the element of sort $\{11,1\}$ is $W_\Sigma(\{x_{0,1}, x_{1,1}\})_1$;

(iv) λ, a, *, + name the constants λ, a, $*y_{0,1}y_{1,0}$, $+ x_{0,1}x_{1,1}$ respectively. $c_{\langle w,v,i\rangle} \in D^{(\Sigma)}_{\langle\langle w,i\rangle\langle v,w_0\rangle\ldots\langle v,w_{n-1}\rangle,\langle v,i\rangle\rangle}$ is assigned the operation of composition described previously. δ_w^{j+1} $(0 \le j \le n-1)$ is assigned the following operation:

If $(t_0,\ldots,t_{n-1}) \in W_\Sigma(X_v)^w$ and $c_{\langle w,v,w_j\rangle} \in$ $D^{(\Sigma)}_{\langle\langle w,w_j\rangle\langle v,w_0\rangle\ldots\langle v,w_{n-1}\rangle,\langle v,w_j\rangle\rangle}$, then $c_{\langle w,v,w_j\rangle}(\delta_w^{j+1}, t_0,\ldots,t_{n-1}) = t_j$. That is, δ_w^{j+1} 'chooses' the (j+1)st element in the list $t_0,\ldots,t_{n-1}$. □

Example 4

Consider the context-free grammar $G = \langle\Sigma,N,P,Z\rangle$ where:

(i) Σ is as in Example 1;

(ii) $N_{\langle\lambda,0\rangle} = \{Z,L\}$, $N_{\langle\lambda,1\rangle} = \{A,C\}$, $N_{\langle 1,1\rangle} = \{B,D\}$, $N_{\langle 10,0\rangle} = \{S\}$;

(iii) $P = \{Z \to S(A,L),\ Z \to S(C,L),\ C \to B(A),\ S(x,y) \to *xy,$
$B(x) \to B(D(x)),\ B(x) \to +xx,\ D(x) \to +xx,\ A \to a,\ L \to \lambda\}$.

Then $L(G) = \{\{*a\lambda, *+aa\lambda, *++aa+aa\lambda, \text{etc.}\}, \phi\}$. If we add Z to $N_{\langle\lambda,1\rangle}$ and $Z \to a$, $Z \to B(A)$ to P (and so get a new grammar G') we get $L(G') = \{\{*a\lambda, *+aa\lambda, *++aa+aa\lambda, \text{etc.}\}, \{a, +aa, ++aa+aa, \text{etc.}\}\}$. □

Example 5

Consider the grammar $G = \langle \Sigma, N, P, Z \rangle$ of Example 4. We will construct the regular grammar $G' = \langle D(\Sigma), D(N), P', Z' \rangle$ as follows:

(i) $D(I) = \{\langle\lambda,0\rangle, \langle\lambda,1\rangle, \langle 1,0\rangle, \langle 1,1\rangle, \langle 10,0\rangle, \langle 10,1\rangle, \langle 11,0\rangle, \langle 11,1\rangle\}$.

(ii) $(D(V))_{\langle\lambda,\langle\lambda,0\rangle\rangle} = \{\lambda, Z', L'\}$;

$(D(V))_{\langle\lambda,\langle\lambda,1\rangle\rangle} = \{a, A', C'\}$;

$(D(V))_{\langle\lambda,\langle 1,0\rangle\rangle} = \phi$;

$(D(V))_{\langle\lambda,\langle 1,1\rangle\rangle} = \{\delta^1_{\langle 1\rangle}, B', D'\}$;

$(D(V))_{\langle\lambda,\langle 10,0\rangle\rangle} = \{\delta^1_{\langle 10\rangle}, *, S'\}$;

$(D(V))_{\langle\lambda,\langle 10,1\rangle\rangle} = \{\delta^1_{\langle 10\rangle}\}$;

$(D(V))_{\langle\lambda,\langle 11,0\rangle\rangle} = \phi$;

$(D(V))_{\langle\lambda,\langle 11,1\rangle\rangle} = \{\delta^1_{\langle 11\rangle}, \delta^2_{\langle 11\rangle}, +\}$;

$(D(V))_{\langle\langle w,i\rangle\langle v,w_0\rangle\ldots\langle v,w_{n-1}\rangle,\langle v,i\rangle\rangle} = \{c_{\langle w,v,i\rangle}\}$

for each $(w,v,i) \in \{1,10,11\} \times \{\lambda,1,10,11\} \times I$.

(iii) P' is: $Z' \to c_{\langle 10,\lambda,0\rangle} S'A'L'$

$Z' \to c_{\langle 10,\lambda,0\rangle} S'C'L'$

$C' \to c_{\langle 1,\lambda,1\rangle} B'A'$

$S' \to c_{\langle 10,10,0\rangle} *\delta^1_{\langle 10\rangle}\delta^2_{\langle 10\rangle}$

$B' \to c_{\langle 1,1,1\rangle} B'D'$

$B' \to c_{\langle 11,1,1\rangle} + \delta^1_{\langle 1\rangle}\delta^1_{\langle 1\rangle}$

$D' \to c_{\langle 11,1,1\rangle} + \delta^1_{\langle 1\rangle}\delta^1_{\langle 1\rangle}$

$A' \to a$

$L' \to \lambda$.

An example of a derivation is:

$$Z' \underset{G'}{\Rightarrow} C_{<10,\lambda,0>}S'A'L'$$

$$\underset{G'}{\Rightarrow} C_{<10,\lambda,0>}C_{<10,10,0>}{}^{*}\delta^{1}_{<10>}\delta^{2}_{<10>}A'L'$$

$$\overset{*}{\underset{G'}{\Rightarrow}} C_{<10,\lambda,0>}C_{<10,10,0>}{}^{*}\delta^{1}_{<10>}\delta^{2}_{<10>}a\lambda (= t).$$

It can be seen that $YIELD_{<\lambda,0>}(t) = {}^{*}a\lambda$. □

Example 6

Consider the alphabet Σ of Example 1. We define the EOL system $G = <\Sigma,N,P,Z>$ as follows:

(i) $N_{<w,i>} = \phi$ for all $<w,i> \in I^{*} \times I$;

(ii) P is: $^{*}xy \rightarrow {}^{*}xy$,

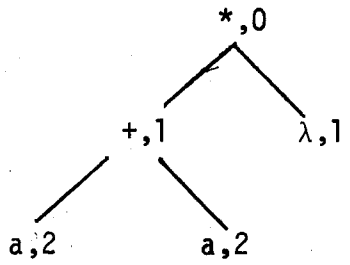
$+xy \rightarrow ++xy+xy$,

$a \rightarrow a$,

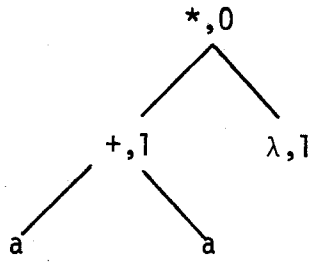
$\lambda \rightarrow \lambda$

(iii) The axiom is $Z = {}^{*}+aa\lambda$.

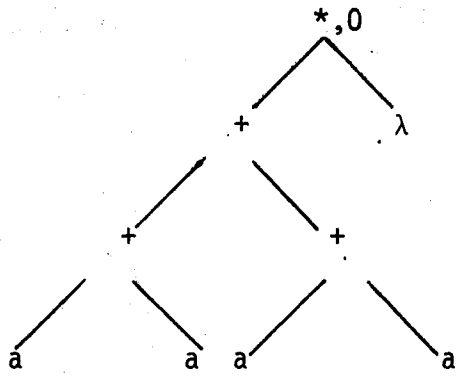
Thus $^{*}+aa\lambda$ is in L(G) (by the reflexivity of $\overset{*}{\underset{G}{\Rightarrow}}$). Consider the following tree corresponding to $^{*}+aa\lambda$ (with the levels of nodes numbered as in the algorithm:



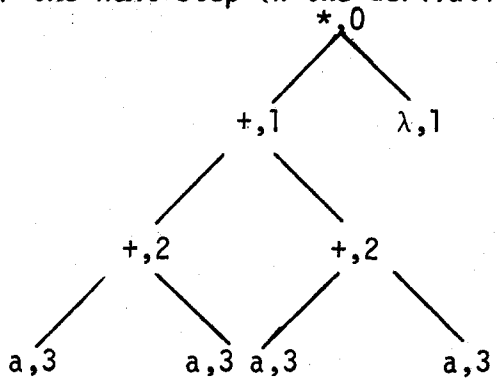
Applying the first step in our algorithm, using the production a → a, we get:



(We drop the label indicating the level when a production has been applied to that node). Now we use +xy → ++xy+xy (with x = y = a) and λ → λ at level 1 to get:

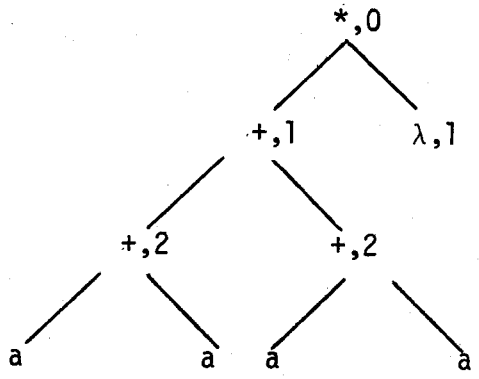


Finally, we apply *xy → *xy to the root to get: (We have renumbered the nodes in readiness for the next step in the derivation).

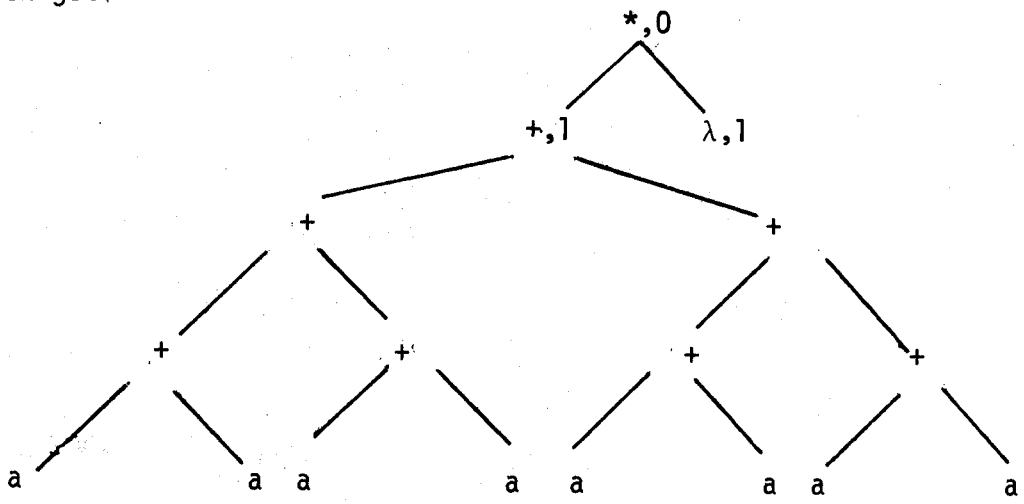


Thus *+aaλ ⇒ *++aa+aaλ.

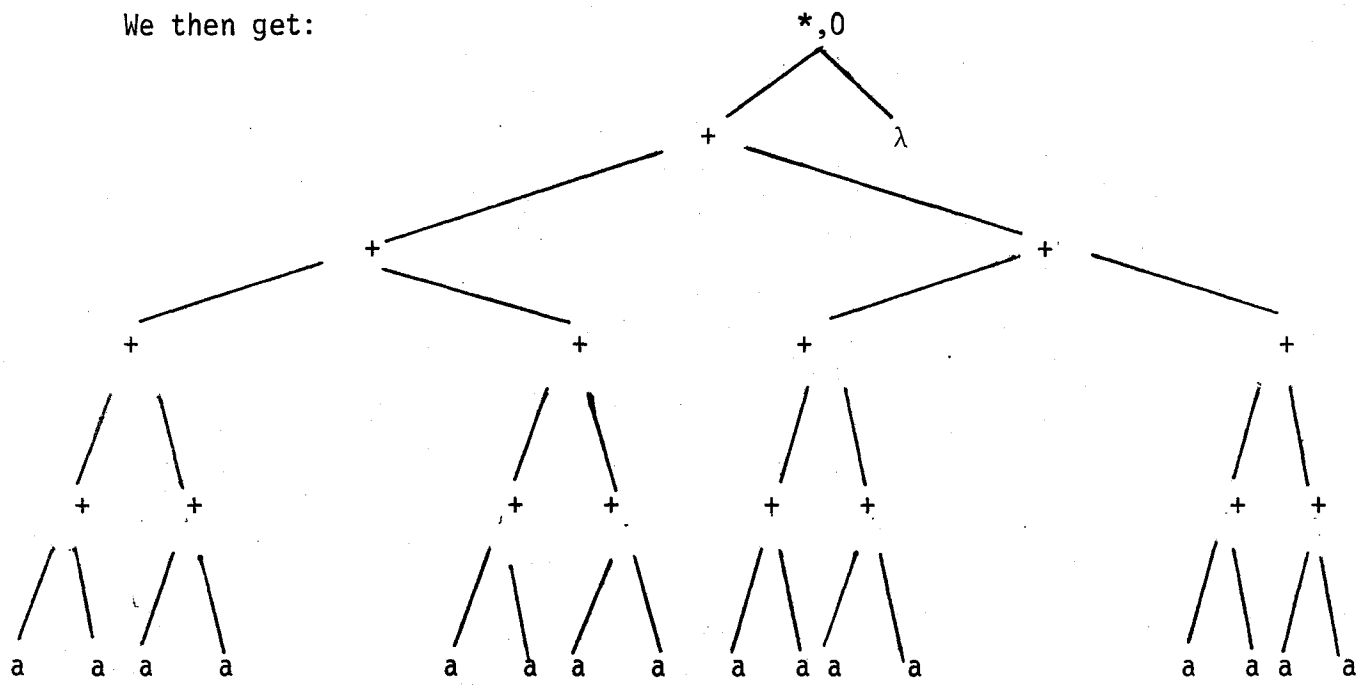
We start again and get at the first step:



We then get:



We then get:



We leave out the last step, as it leaves the tree unchanged. Thus $*{+}{+}aa{+}aa\lambda \underset{G}{\Rightarrow} *{+}{+}{+}{+}aa{+}aa{+}{+}aa{+}aa{+}{+}{+}aa{+}aa{+}{+}aa{+}aa\lambda$. The YIELD of these expressions (considering Σ to be the derived alphabet of the string alphabet $\{a\}$ (with λ the empty string)) is the set $\{a^{2^{2^n}} \mid n \geq 0\}$.

Note that G is actually an OL system. □

Example 7

Let $G = \langle \Sigma, N, F, P, Z \rangle$ where

(i) V is a one-sorted (i.e. ranked) alphabet such that $\Sigma_0 = \{a\}$, $\Sigma_2 = \{+\}$, $N_0 = \{Z\}$, $N_1 = \{B\}$;

(ii) $F = \{f,g\}$ where

$f = \{B(x) \rightarrow B(B(x))\}$

and $g = \{B(x) \rightarrow +xx\}$;

(iii) $P = \{S \rightarrow (g,B)(a), B(x) \rightarrow (f,B(x))\}$.

$$Z \underset{G}{\Rightarrow} (g,B)(a)$$

$$\overset{*}{\underset{G}{\Rightarrow}} (gff,B)(a)$$

$$\underset{G}{\Rightarrow} (gf,B)((gf,B)(a))$$

$$\underset{G}{\Rightarrow} (gf,B)((g,B)((g,B)(a)))$$

$$\overset{*}{\underset{G}{\Rightarrow}} (gf,B)(++aa+aa)$$

$$\underset{G}{\Rightarrow} (g,B)((g,B)(++aa+aa))$$

$$\underset{G}{\Rightarrow} (g,B)(+++aa+aa++aa+aa)$$

$$\underset{G}{\Rightarrow} \; ++++aa+aa++aa+aa+++aa+aa++aa+aa.$$

$Y(L(G))$ is seen to be $\{a^{2^{2^n}} \mid n \geq 0\}$ which is not an indexed set of strings). □

REFERENCE

[1] A.V. Aho, Indexed Grammars - An Extension of Context Free Grammars, Journal of the ACM, 1968, V.15, 647-671.

[2] K. Culik II and J. Opatrný, Macro OL system, Research Report #CS-73-06, Department of Applied Analysis and Computer Science, University of Waterloo.

[3] A. Lindenmayer, Mathematical models for cellular interactions in development, Parts I and II, Journal of Theoretical Biology, 1968, V.18, 280-315.

[4] A. Lindenmayer, Developmental systems without cellular interactions, their languages and grammars, Journal of Theoretical Biology, 1971, V.30, 455-484.

[5] T.S.E. Maibaum, The characterisation of the derivation trees of context-free sets of terms as regular sets, Proc. 13th IEEE Symp. on Switching and Automata Theory, 1972, 224-230.

[6] T.S.E. Maibaum, A generalised approach to formal languages, to appear in Journal of Computer and System Sciences.

[7] T.S.E. Maibaum, Generalised Grammars and homomorphic images of recognizable sets, Doctoral Dissertation, University of London, 1973.

[8] T.S.E. Maibaum, Indexed grammars on terms and their generalisation, in preparation.

[9] J. Opatrný, in preparation.

[10] G. Rozenberg, TOL systems and languages, to appear in Information and Control.

[11] G. Rozenberg, Extension of tabled OL systems and languages, to appear in International Journal of Computer and Information Sciences.

[12] R. Turner, Doctoral Dissertation, University of London, 1973.