International Journal of Computer and Information Sciences, Vol. 3, No. 4, 1974

Some Restrictions on W-Grammars

Sheila A. Greibach[1]

Received December 1973; revised February 1974

The effect of some restrictions on W-grammars (the formalization of the syntax of ALGOL 68) are explored. Two incomparable families examined at length are W_{RB} (languages generated by normal regular-based W-grammars) and W_S (languages generated by simple W-grammars). Both properly contain the context-free languages and are properly contained in the family of quasi-realtime languages. In addition, W_{RB} is closed under nested iterated substitution (but is not an *AFL*) and is properly contained in the family of index languages.

1. INTRODUCTION

One of the reasons for the early and still continuing popularity of context-free grammars has been their use in the formal definition of parts of the syntax of ALGOL and similar programming languages[19] and their role in syntax-directed compilation (cf. Ref. 3 for a discussion of the relationship of context-free grammars and programming languages). However, it was soon found that context-free grammars were not altogether satisfactory models for programming language structure, both because some constructs either could not be defined altogether or could not be conveniently defined by context-free systems, and because such grammar representations did not cover the connections between syntax and semantics. Many attempts have been made on various levels to fill the gaps—indexed languages,[1] macrogrammars,[11,12] and property grammars[23] are only a few examples that come to mind.

The syntax of ALGOL 68[24] provided an altogether different departure. The concept of a context-free grammar was generalized to allow an infinite

The research represented in this paper was supported in part by the National Science Foundation under Grant GJ-803.

[1] Department of System Science, University of California, Los Angeles, California.

but recursive set of context-free productions. The productions arise from the interaction of two context-free grammars. The first grammar feeds to the second grammar the names of nonterminals in the second grammar. The rules of the second grammar are really rule schemas with place holders which we call metavariables. When each metavariable is uniformly replaced by a string generated in the first level grammar the result is a production then applied in standard context-free fashion. For a discussion of the reasons for such a system and the utility of double-level grammars in programming language definition the reader is referred to Cleaveland and Uzgalis.[10]

The type of generating system introduced in the ALGOL 68 draft[24] has been variously called "the syntax of ALGOL 68," "double level grammar," "van Wijngarden syntax," and "*W*-grammar." We select the term "*W*-grammar" as being both suggestive and simple.

The concept of a *W*-grammar has been formalized by Sintzoff,[22] Chastellier and Colmerauer,[9] Baker,[4] and others. The various definitions differ in details though the underlying notion is identical. We introduce below a definition adapted from that of Cleaveland and Uzgalis[10] and modified to—hopefully—be cleaner and clearer for reader and typist alike; the essential idea is unchanged.

It was quickly noticed that *W*-grammars define the family of recursively enumerable sets,[22] while suitable restrictions yield recursive sets[17] or context-sensitive languages.[4] Thus the utility and power of *W*-grammars are unsurprising. However, it does not appear necessary to use the full power of *W*-grammars in language definition. Thus it seems reasonable to study restrictions on *W*-grammars—besides those mentioned above—which yield recursive languages and hopefully combine descriptive power with attractive mathematical properties (ease of recognition, decidable emptiness question, interesting closure properties, etc.).

In this paper we define two restrictions on *W*-grammars and study their properties, interrelationship, and connections with other grammars and machines. The first restriction is to normal and regular-based grammars. "Normal" means, roughly, that the left-hand side of a second-level rule schema consists of one metavariable and so the rule schemas themselves appear "context-free." Regular-based means that the first-level grammar is finite-state. It is easy to see that either restriction by itself still permits the generation of all recursively enumerable sets. Together, however, the restrictions define a family W_{RB} of languages properly containing the context-free languages and properly contained in the family of indexed languages and hence a fortiori in the context-sensitive languages. Membership, emptiness, and finiteness are decidable for normal regular-based *W*-grammars. Languages in W_{RB} can be recognized by nondeterministic multitape Turing machines in realtime. The closure properties of W_{RB} are unusual—W_{RB} is closed under nonerasing substitution, in fact, under nonerasing nested iterated substitution, but not under intersection with regular sets.

The other family investigated is the family W_S of languages generated by *simple* W-grammars. In a simple grammar each nonterminal in the infinite production set arises from an individual metavariable and only terminal strings replace metavariables in the rule schemes of the second-level grammar. Like W_{RB}, W_S properly contains the family of context-free languages and is properly contained in the family of languages accepted by nondeterministic multitape Turing machines. Unlike W_{RB}, W_S has no interesting closure properties. The families W_{RB} and W_S are incomparable—neither is contained in the other. Curiously enough, although members of W_S are recursive, membership is undecidable for simple W-grammars. Every language expressible as the intersection of *two* context-free languages is in W_S, but some languages expressible as the intersection of *three* context-free languages are not.

2. BASIC CONCEPTS

In this section we introduce the basic definitions, notation, and concepts we shall use and outline briefly our specific results. We assume that the reader is familiar with the notions of context-free grammars, derivation trees, and finite-state machines and is acquainted with the basic properties of context-free languages and regular sets (for background material see Hopcroft and Ullman[15] or Aho and Ullman[3]).

In order to avoid uninteresting and trivial variations to handle the empty word (designated by e), we shall assume that all languages are "e-free," that is, do not contain the empty word. Thus by regular, context-free, or context-sensitive language, we understand e-free regular, context-free, or context-sensitive language. Most of the results do hold if the empty word is added by ad hoc methods; the interested reader can easily work out the exceptions for the empty word.

Notation. For a language L we let $L^+ = \{w_1 \cdots w_n \mid n \geqslant 1, w_i \in L\}$ and $L^* = L \cup \{e\}$. If L is finite, then $|L|$ is the number of elements in L. For a word w we let $|w|$ be the length of w.

We designate a *grammar* by $G = (V, \Sigma, P, \sigma)$, where V is a finite vocabulary, $\Sigma \subset V$, $\sigma \in V - \Sigma$, and P is a finite set of *rules* or *productions* of the form $u \rightarrow v$, $u \in (V - \Sigma)^+$, $v \in V^*$. We call σ the *start symbol*, members of Σ *terminals*, and members of $V - \Sigma$ *nonterminals*, *intermediates*, or *variables*. If $u \rightarrow v$ is in P, $x, y \in V^*$, then $xuy \Rightarrow_G xvy$. Then $\Rightarrow_G^*$ is the transitive reflexive extension of $\Rightarrow_G$; we omit the subscript G when no confusion can occur. The *language generated by* G is $L(G) = \{w \in \Sigma^+ \mid \sigma \Rightarrow^* w\}$. If $|u| \leqslant |v|$ for each $u \to v$ in P, then G is *context-sensitive*. If $u \in V - \Sigma$ whenever $u \to v$ is in P, then G is *context-free*. If each rule in P is of the form $Z \to wY$ or $Z \to u$ for $Z, Y \in V - \Sigma$, $w \in \Sigma^*$, $u \in \Sigma^+$, then G is *regular* or *finite-state*. If G is context-sensitive (context-free, regular), then $L(G)$ is a *context-sensitive* (*context-free*, *regular*) *language*.

We are now ready to introduce our notation for W-grammars.

Definition 2.1. *A W-grammar* $G = (V_M, V_P, \Sigma, P_M, P_h, \sigma)$ consists of a finite set V_M of *metavariables*, a finite set V_P partitioned into *terminals* Σ and *protovariables* $V_P - \Sigma$, a finite set P_M of *metaproductions*, a finite set P_h of *hyperrules*, and a *start symbol* $\sigma \in V_P - \Sigma$ such that: (1) $V_P \cap V_M = \varnothing$ and "$<$" and "$>$" do not appear in $V_P \cup V_M$, (2) for each $A \in V_M$, $G_A = (V_M \cup V_P, V_P, P_M, A)$ is a context-free grammar, and (3) each hyperrule in P_h is of the form $Z \to y$ for $Z \in H \cup (V_P - \Sigma)$ and $y \in (V_M \cup V_P \cup H)^+$ where $H = \{\langle\alpha\rangle \mid \alpha \in (V_M \cup V_P)^+\}$. We call H the set of *hypernotions*.

If α is a string of terminals, protovariables, and metavariables, then $\langle\alpha\rangle$ is a hypernotion. A hypernotion $\langle\alpha\rangle$ is a place holder for one or more "variables" $\langle w\rangle$ of the second-level grammar. Hypernotions and metavariables are replaced in the rules of P_h as follows. For $A \in V_M$, $G_A = (V_M \cup V_P, V_P, P_M, A)$ is a *metagrammar* of G; if $A \Rightarrow^* w$ in G_A, we say *A metagenerates w*. We let $L_A = L(G_A)$. A *metaassignment* of $V_P \cup V_M$ is a homomorphism h such that $h(<) = <$, $h(>) = >$, $h(Z) = Z$ for $Z \in V_P$, and $h(A) \in L_A$ for $A \in V_M$. If $Z \to y$ is a hyperrule and h a metaassignment, then $h(Z) \to h(y)$ is a production of G. We let P be the set of all such productions; the production set may be infinite.

We can define derivations of G in two ways. First, let $I = \{\langle\alpha\rangle \mid \alpha \in V_P^+\} \subseteq H$. If I' is a finite subset of I, we can regard I' as a set of distinct nonterminals. If P' is a finite subset of P, and $G' = (I' \cup V_P, \Sigma, P', \sigma)$ happens to be a context-free grammar, then G' is a subgrammar of G; if $w_1 \Rightarrow_{G'}^* w_2$, then $w_1 \Rightarrow_G^* w_2$. The *language generated by* G is $L(G) = \bigcup_{\text{subgrammar } G'} L(G')$. We call $L(G)$ a *W-language*.

Alternatively, derivations of G can be defined directly from P. If $w_1, w_2 \in (V_P \cup I)^*$, $\alpha \in V_P^+$, and $\langle\alpha\rangle \to y$ is in P, then $w_1\langle\alpha\rangle w_2 \Rightarrow_G w_1 y w_2$. If $Z \in V_P - \Sigma$ and $Z \to y$ is in P, then $w_1 Z w_2 \Rightarrow w_1 y w_2$. As usual, $\Rightarrow_G^*$ is in the transitive reflexive extension of $\Rightarrow_G$. Further, if we have a derivation

$$\gamma\colon \quad \sigma \underset{G}{\Rightarrow} w_1 \underset{G}{\Rightarrow} \cdots \underset{G}{\Rightarrow} w_n \quad \text{for} \quad w_n \in \Sigma^*$$

then γ is a *complete derivation in G*. Then

$$L(G) = \{w \in \Sigma^* \mid \sigma \overset{*}{\underset{G}{\Rightarrow}} w\}$$

Let us illustrate these rather complicated definitions by some examples which shall be useful in the sequel.

Example 1. A W-grammar for $\{a^{2^n} \mid n \geqslant 1\}$. There is one metavariable, N, one protovariable, σ, and one terminal, a. The metaproductions are $N \to aN$ and $N \to a$, and the hyperrules are $\sigma \to \langle a \rangle$, $\langle N \rangle \to \langle NN \rangle$, and $\langle N \rangle \to N$.

Notice that $L_N = a^+$. Any metaassignment h is of the form $h(N) = a^n$, $n \geqslant 1$. Thus

$$P = \{\langle a^n \rangle \to \langle a^{2n} \rangle \mid n \geqslant 1\} \cup \{\langle a^n \rangle \to a^n \mid n \geqslant 1\} \cup \{\sigma \to \langle a \rangle\}$$

Thus the only complete derivations are of the forms:

$$\sigma \Rightarrow \langle a \rangle \Rightarrow \langle aa \rangle \Rightarrow \langle aaaa \rangle \Rightarrow \cdots \Rightarrow \langle a^{2^n} \rangle \Rightarrow a^{2^n}$$

Example 2. A W-grammar for $\{a^{n^2} \mid n \geqslant 1\}$. We have $V_M = \{N\}$, $V_P = \{a, \sigma\}$, $P_M = \{N \to a, N \to NN\}$ and

$$P_h = \{\sigma \to \langle a \rangle, \langle N \rangle \to N\langle aaN \rangle, \langle N \rangle \to N\}$$

Thus whenever $\langle a^n \rangle$ actually appears in a complete derivation of G, n is odd. Complete derivations of G proceed:

$$\begin{aligned} \sigma &\Rightarrow \langle a \rangle \Rightarrow a\langle a^3 \rangle \Rightarrow aa^3\langle a^5 \rangle \Rightarrow \cdots \\ &\Rightarrow aa^3 \cdots a^{2k-1}\langle a^{2k+1} \rangle \Rightarrow aa^3 \cdots a^{2k-1}a^{2k+1} \end{aligned}$$

for $k \geqslant 0$. Since

$$\sum_{m=0}^{k} a^{2m+1} = a^{(k+1)^2}$$

the grammar generates the desired language.

Example 3. A W-grammar for $\{a^n \mid n \geqslant 4, n \text{ is not prime}\}$. Let

$$V_M = \{N\}, \qquad V_P = \{a, \sigma\}, \qquad P_M = \{N \to aa, N \to aN\},$$

and

$$P_h = \{\sigma \to \langle N \rangle, \langle N \rangle \to N\langle N \rangle, \langle N \rangle \to NN\}$$

The productions form

$$\begin{aligned} P = \{\sigma \Rightarrow \langle a^n \rangle \mid n \geqslant 2\} &\cup \{\langle a^n \rangle \to a^n\langle a^n \rangle \mid n \geqslant 2\} \\ &\cup \{\langle a^n \rangle \to a^n a^n \mid n \geqslant 2\} \end{aligned}$$

The start set is $L_N = \{a^n \mid n \geqslant 2\}$. Thus derivations proceed, for $n \geqslant 2$:

$$\sigma \Rightarrow \langle a^n \rangle \Rightarrow a^n \langle a^n \rangle \Rightarrow a^n a^n \langle a^n \rangle \Rightarrow \cdots \Rightarrow (a^n)^m \langle a^n \rangle \Rightarrow a^{n(m+2)}$$

So G defines

$$\{a^{n(m+2)} \mid n \geqslant 2, m \geqslant 0\}$$

which is another way of expressing the desired language.

Example 4. Some W-grammars for $L_1 = \{a_1{}^n a_2{}^n a_3{}^n \mid n \geqslant 1\}$ and $L_2 = \{a_1{}^n a_2{}^n a_3{}^n a_4{}^n \mid n \geqslant 1\}$. We have in both cases $V_M = \{A\}$, $\Sigma = \{a_1, a_2, a_3, a_4\}$, $V_P = \Sigma \cup \{\sigma\}$, and

$$P_M = \{A \to a_2 A, A \to a_3 A, A \to a_2, A \to a_3\}$$

For the first language we have

$$P_{h,1} = \{\sigma \to \langle a_2 a_3 \rangle, \langle A \rangle \to a_1 \langle a_2 A a_3 \rangle, \langle A \rangle \to a_1 A\}$$

and in the second case

$$P_{h,2} = \{\sigma \to \langle a_2 a_3 \rangle, \langle A \rangle \to a_1 \langle a_2 A a_3 \rangle a_4, A \to a_1 A a_4\}$$

Notice that both grammars have the properties: (1) The metagrammars are regular, and (2) whenever $\langle \alpha \rangle$ appears on the left-hand side of a hyperrule, $\alpha \in V_M$ (in this case $\alpha = A$). We shall see in the proof of Theorems 5.1 and 5.2 that any W-grammars for L_1 and L_2 with properties (1) and (2) more or less "look like" the ones constructed here. Further, if we add a fifth coordinate a_5, we must relax either (1) or (2) as is done in the next example.

Example 5. Two W-grammars for $L = \{a_1{}^n a_2{}^n \cdots a_5{}^n \mid n \geqslant 1\}$.

1. We first construct a W-grammar whose metagrammar is not regular. Let $V_M = \{A, B, A_1, A_2, A_3, A_4\}$, $\Sigma = \{a_1, a_2, a_3, a_4, a_5\}$, $V_P = \Sigma \cup \{\sigma\}$,

$$\begin{aligned} P_M = \{&A \to A_1 A_3 a_5, A \to A a_5, A_1 \to a_1 A_1 a_2, A_3 \to a_3 A_3 a_4, \\ &A_1 \to a_1 a_2, A_3 \to a_3 a_4, B \to a_1 B, B \to a_1 A_2 A_4, \\ &A_2 \to a_2 A_2 a_3, A_4 \to a_4 A_4 a_5, A_2 \to a_2 a_3, A_4 \to a_4 a_5\} \end{aligned}$$

and

$$P_h = \{\sigma \to \langle A \rangle, \langle B \rangle \to B\}$$

Notice that

$$L_A = \{a_1{}^n a_2{}^n a_3{}^m a_4{}^m a_5{}^k \mid n, m, k \geqslant 1\}$$

and

$$L_B = \{a_1{}^k a_2{}^n a_3{}^n a_4{}^m a_5{}^m \mid n, m, k \geqslant 1\}$$

and the only completed derivations are

$$\sigma \Rightarrow \langle w \rangle \Rightarrow w$$

for w in $L_A \cap L_B$. Since

$$L = L_A \cap L_B$$

the W-grammar generates L.

2. We now give a W-grammar for L with a regular metagrammar. Let $V_M = \{N\}$, $V_P = \Sigma \cup \{\sigma, a\}$, $P_M = \{N \to a, N \to aN\}$, and

$$\begin{aligned} P_h = &\{\sigma \to a_1a_2a_3a_4a_5\} \cup \{\sigma \to \langle a_1N\rangle\langle a_2N\rangle\langle a_3N\rangle\langle a_4N\rangle\langle a_5N\rangle\} \\ &\cup \{\langle a_iaN\rangle \to a_i\langle a_iN\rangle \mid i = 1, 2, 3, 4, 5\} \\ &\cup \{\langle a_ia\rangle \to a_ia_i \mid i = 1, 2, 3, 4, 5\} \\ &\cup \{\langle a_i\rangle \to a_i \mid i = 1, 2, 3, 4, 5\} \end{aligned}$$

It is quite apparent that the family of languages generated by W-grammars is the family of recursively enumerable languages, and that remains true even if we impose some strong-appearing restrictions.

Definition 2.2. Let $G = (V_M, V_P, \Sigma, P_M, P_h, \sigma)$ be a W-grammar. We call G *regular-based* if for each $A \in V_M$, $G_A = (V_M \cup V_P, V_P, P_M, A)$ is regular. We call G *normal* if each hyperrule is of the form $Z \to y$ or $\langle A\rangle \to y$ for $Z \in V_P - \Sigma$, $A \in V_M$, $y \in (V_M \cup V_P \cup H)^+$. We call G *unary* if each hyperrule is of the form $Z \to y$ or $\langle A\rangle \to y$ for $Z \in V_P - \Sigma$, $A \in V_M$, $y \in (V_M \cup V_P \cup \{\langle B\rangle \mid B \in V_M\})^+$. We call G *strict* if for each $A \in V_M$, $L_A \subseteq \Sigma^+$.

Definition 2.3. A W-grammar $G = (V_M, V_P, \Sigma, P_M, P_h, \sigma)$ is *lossless* if for each hyperrule of the form

$$\langle \alpha\rangle \to u_0\langle \beta_1\rangle\, u_1\langle \beta_2\rangle \cdots u_{n-1}\langle \beta_n\rangle u_n$$

for $n \geqslant 0$ (if $n = 0$, the rule is $\langle\alpha\rangle \to u_0$), $\alpha, \beta_i \in (V_M \cup V_P)^+$, $1 \leqslant i \leqslant n$, $u_i \in (V_M \cup V_P)^*$, $0 \geqslant i \geqslant n$, we have

$$|\,\alpha\,| \leqslant |\,u_0\beta_1 \cdots u_{n-1}\beta_n u_n\,|$$

and for each $A \in V_M$

$$\#_A(\alpha) \leqslant \#_A(u_0\beta_1 \cdots u_{n-1}\beta_n u_n)$$

where $\#_A(w)$ is the number of occurrences of A in w.

Suppose $G = (V, \Sigma, P, S)$ is a grammar. If we construct three metavariables A, B, and T with $L_A = L_B = V^+$ and $L_T = \Sigma^+$, and for each rule $u_i \to v_i$ in P four hyperrules $\langle Au_iB\rangle \to \langle Av_iB\rangle$, $\langle Au_i\rangle \to \langle Av_i\rangle$, $\langle u_iA\rangle \to \langle v_iA\rangle$, and $\langle u_i\rangle \to \langle v_i\rangle$ and then add hyperrules $\sigma \to \langle S\rangle$ and $\langle T\rangle \to T$, we clearly have a W-grammar generating $L(G)$. Thus the result of Sintzoff[22] can be expressed as follows.

Theorem 2.1. (Sintzoff). Every recursively enumerable language can be generated by a strict regular-based W-grammar.

An alternative construction shows that normal W-grammars also generate all recursively enumerable languages.

Theorem 2.2. Every recursively enumerable language can be defined by a strict normal W-grammar.

Proof. Let L be a recursively enumerable language. Then L can be expressed as the homomorphic image of the intersection of two context-free languages.[13] Further, if $L \subseteq \Sigma^+$, one can assume that there is a finite vocabulary Δ and context-free languages L_1 and L_2 such that $\Sigma \cap \Delta = \varnothing$, $L_1 \cup L_2 \subseteq \Sigma^+\Delta^+$, and $L = \{w \in \Sigma^+ \mid \exists x \in \Delta^+, wx \in L_1 \cap L_2\}$. That is, L can be obtained from $L_1 \cap L_2$ by chopping off the "tail" in Δ^+.

Let σ, D, d, A, S_1, and S_2 be new symbols, and for each a in Σ let $\bar{a}$ be new. The symbols D, A, S_1, S_2, and $\bar{a}$ will be metavariables. Since L_1 and L_2d are context-free, we can add metavariables and metaproductions so that $L_D = \Delta^+$, $L_A = \Sigma^+$, $L_{S_1} = L_1$, $L_{S_2} = L_2d$, and $L_{\bar{a}} = \Sigma^*a\Delta^+dd$ for $a \in \Sigma$.

The hyperrules are

$$\sigma \to A\langle AaD\rangle, \qquad \sigma \to \langle aD\rangle, \qquad \langle \bar{a}\rangle \to a, \qquad a \in \Sigma$$
$$\langle S_1\rangle \to \langle S_1d\rangle, \qquad \langle S_2\rangle \to \langle S_2d\rangle$$

Thus the only complete derivations are of the form:

$$\sigma \Rightarrow w\langle waz\rangle \Rightarrow w\langle wazd\rangle \Rightarrow w\langle wazdd\rangle \Rightarrow wa$$

for $w \in \Sigma^*$, $a \in \Sigma$, $z \in \Delta^+$, $waz \in L_1 \cap L_2$ and so $wa \in L$. $\square$

Thus one must consider carefully whether there are any sets of restrictions on a W-grammar G such that $L(G)$ is guaranteed to be recursive and yet G still has considerably more expressive power than a context-free grammar. One obvious possibility is to examine lossless W-grammars. If G is lossless, every production in P is nondecreasing except for possible elimination of $\langle$ and $\rangle$; by inventing new symbols we surely can encode $\langle w\rangle$ within $|w|$ steps without undue complication. Further, if $\langle\beta\rangle$ appears in a hyperrule, the language

$$L_\beta = \{h(\beta) \mid h \text{ metaassignment}\}$$

is certainly context-sensitive since each L_A is context-free and context-sensitive grammars can duplicate. Thus a multitape Turing machine can certainly obtain a production p from a hyperrule, and apply p to obtain $w_1 \Rightarrow_G w_2$, using no more than $\text{Max}(|w_1|, |w_2|) = |w_2|$ tape squares. It is well known that any rewriting system such that each derivation step is nondecreasing and there is a multitape Turing machine which can imitate each step $w_1 \Rightarrow w_2$ using no more than $|w_2|$ squares yields a context-sensitive language.

The details of such a construction for lossless W-grammars is given by Baker,[4] who shows the following.

Theorem 2.3. (Baker). The family of languages generated by lossless W-grammars is precisely the family of context-sensitive languages.

In this paper we shall examine two possible restrictions on W-grammars which yield proper (and incomparable) subclasses of the family of context-sensitive languages while still retaining some of the W-grammar's facility for duplicating and comparing substrings.

We observed that neither the requirement to be regular-based nor that to be normal restricts the generative power of a W-grammar, although they may very well restrict the ease or naturalness of producing certain structures. On the other hand, restricting a W-grammar to be *both* regular-based *and* normal is a significant restriction on the generative capacity.

In Sections 3–5 we study W_{RB}, the family of languages generated by normal regular-based W-grammars. The key result (Theorem 3.1) is that every normal regular-based W-grammar can be effectively converted into a lossless normal regular-based W-grammar in a special factored form. In Theorem 3.2 we prove a reduced form theorem similar to the one for context-free grammars.[5] Using this result, we show that although Examples 1–4 show that normal regular-based W-grammars have considerable facility for computing numerical functions, duplicating strings, and comparing numbers, many questions—notably membership, emptiness, and finiteness—are decidable for such grammars (Theorem 3.3).

In Theorem 4.1 we show that W_{RB} is contained in the family of indexed languages[1] (which is also the family of nested stack languages[2] and of OI macrolanguages[11,12]). Thus W_{RB} is obviously a proper subset of the family of context-sensitive languages. More than that, each member of W_{RB} can be accepted in realtime by a nondeterministic multitape Turing machine (Theorem 4.2).

The containments in Theorems 4.1 and 4.2 are proper—the language of Example 5, for example, is not in W_{RB} (Theorem 5.2). Finally, we conclude Section 5 by noticing that W_{RB} has curious closure properties—it is closed under nested iterated substitution but not under intersection with regular sets (Theorem 5.3).

The other subfamily of W-grammars that we shall study is the family of strict unary W-grammars. We shall call a strict unary W-grammar *simple* and let W_S be the family of languages accepted by simple W-grammars. The family W_S is a very strange one. On one hand, simple W-grammars can do with ease many things beyond the power of normal regular-based grammars—e.g., W_S contains every language expressible as the intersection of *two* context-free languages (Theorem 6.1); Example 5 gives a language in $W_S - W_{RB}$. On the other hand, Examples 1 and 2 give examples of languages in $W_{RB} - W_S$; if f is a monotone increasing function from positive integer to positive integers, $\{a^{f(n)} \mid n \geqslant 1\}$ cannot be in W_S if f grows more than linearly (cf. Theorem 6.4). Also, there are languages expressible as the intersection of *three* context-free languages which are not in W_S (Theorem 6.5). Now W_S is, like W_{RB}, a proper subset of $\mathscr{Q}$, the family of languages accepted in realtime by nondeterministic multitape Turing machines. *But* this inclusion is nonconstructive—there is no algorithm to transform a simple W-grammar into an equivalent context-sensitive grammar, although one always exists! For example, membership is not decidable for simple W-grammars (Theorem 6.2), although each member of W_S is recursive! There are many pathological systems behaving in this manner, but this is one of the few examples of more or less "natural" systems exhibiting this sort of behavior. The closure properties of W_S are uninteresting—W_S is closed under almost none of the natural operations on languages (except for Kleene $+$).

Finally, in Section 7 we mention some open problems regarding the exact relationship between W_S and $\mathscr{Q}$, as well as introduce some other restrictions and extensions W-grammars it might be profitable to study.

3. NORMAL REGULAR-BASED W-GRAMMARS

As we saw, regular-based W-grammars have the same generative power as context-free based W-grammars, so there is little point in studying them. We shall instead focus attention on normal regular-based W-grammars, which have considerable power, as Examples 1–4 show, and yet have many pleasant properties.

Definition 3.1. A W-grammar $G = (V_M, V_P, \Sigma, P_M, P_h, \sigma)$ is *regular* if it is normal and regular-based and all the rules are of the forms $Z \to w_1 Y$, $Z \to w_1\langle y\rangle$, $\langle A\rangle \to w_1 Y$, $\langle A\rangle \to w_1\langle y\rangle$, $Z \to w_2$, and $\langle A\rangle \to w_2$ for $Y, Z \in V_P - \Sigma$, $A \in V_M$, $w_1 \in (V_M \cup \Sigma)^*$, and $w_2, y \in (V_M \cup \Sigma)^+$. Let

$$W_{RB} = \{L(G) \mid G \text{ normal and regular-based}\}$$

and

$$W_R = \{L(G) \mid G \text{ regular}\}$$

The following should be immediately apparent.

Corollary. The family of context-free languages is properly contained in W_{RB}.

Most of our results will depend on converting a normal regular-based grammar into a special form.

First, we extend the notation L_A, for a metavariable A, to L_β for β composed of metavariables, protovariables, and terminals.

Definition 3.2. Let $G = (V_M, V_P, \Sigma, P_M, P_h, \sigma)$ be a W-grammar. For $\beta \in (V_M \cup V_P)^+$ let

$$L_\beta = \{h(\beta) \mid h \text{ is a metaassignment}\}$$

Thus for $A \in V_P$, $L_A = \{A\}$, and if $A_1, \ldots, A_n \in V_M \cup V_P$, then $L_{A_1 \cdots A_N}$ is the collection of all words $w_1 \cdots w_n$ such that $w_i \in L_{A_i}$ and whenever $A_i = A_j$, then $w_i = w_j$.

Now we define a factored form for normal W-grammars.

Definition 3.3. A normal W-grammar $G = (V_M, V_P, \Sigma, P_M, P_h, \sigma)$ is *factored* if (1) for all $A \in V_M$, $L_A \neq \varnothing$; (2) for all $A, B \in V_M$ either $L_A = L_B$ or $L_A \cap L_B = \varnothing$; and (3) if $\langle\beta\rangle$ appears in any hyperrule, there is an $A \in V_M$ such that $L_\beta \subseteq L_A$.

Now we shall see that every normal regular-based W-grammar can be converted into a lossless factored one. This is an easy consequence of Nerode's[20] theorem for regular sets. Recall that for a finite vocabulary T^* an equivalence relation $\sim$ on T^* is a *congruence* relation if whenever $u \sim v$ and $x \sim y$, then $ux \sim vy$; it is of *finite index* if T^* is partitioned by $\sim$ into a finite number of equivalence classes.

Theorem 3.1. Given a normal regular-based grammar G, we can construct a lossless, factored, normal, regular-based grammar $\bar{G}$ such that $L(G) = L(\bar{G})$, and $\bar{G}$ is regular if G is regular, strict if G is strict, and unary if G is unary.

Proof. Let $G = (V_M, V_P, \Sigma, P_M, P_h, \sigma)$. We know that each L_A is regular for $A \in V_M$. Hence there is a congruence relation of finite index on V_P^*, call it $\sim$, such that each L_A is the union of some of the congruence classes of $\sim$ on V_P^*.[20] Let $\mathscr{E}$ be the set of these congruence classes. For $A \in V_M$ let $\mathscr{E}_A = \{E \in \mathscr{E} \mid E \cap L_A \neq \varnothing\}$; thus $L_A = \bigcup_{E \in \mathscr{E}_A} E$. Now since $\sim$ is a congruence relation, given E_1 and E_2 in $\mathscr{E}$, there is a unique E_3 in $\mathscr{E}$ such that $E_1 E_2 = \{ux \mid u \in E_1, x \in E_2\} \subseteq E_3$. Let us write for convenience

$$E_1 \cdot E_2 = E_3$$

For $u \in V_P^*$, let $[u]$ be the equivalence class of u. For each $A \in V_M$, $E \in \mathscr{E}_A$, let $\hat{E}$, $\bar{E}$, and (E, A) be new symbols. Thus if $A, B \in V_M$, $A \neq B$, and $E \in \mathscr{E}_A \cap \mathscr{E}_B$, (E, A) and (E, B) will be distinct symbols—two different metavariables.

Each equivalence class E in $\mathscr{E}$ is a regular set and so $E - \{e\}$ is regular. Thus we can add metavariables to form a set $\bar{V}_M$, containing $\{\bar{E} \mid E \in \mathscr{E}\} \cup \{(E, A) \mid A \in V_M, E \in \mathscr{E}_A\}$, and a set $\bar{P}_M$ of metaproductions such that for $E \in \mathscr{E}$, $L_{\bar{E}} = E - \{e\}$ and for $A \in V_M$, $E \in \mathscr{E}_A$, $L_{(E,A)} = L_{\bar{E}}$, and our ultimate W-grammar G will be regular-based.

The set $\bar{V}_P$ of protovariables and terminals for $\bar{G}$ is given by

$$\bar{V}_P = V_P \cup \{\hat{E} \mid E \in \mathscr{E}\}$$

Each $(\bar{V}_M \cup \bar{V}_P, \bar{V}_P, \bar{P}_M, E)$ or $(\bar{V}_M \cup \bar{V}_P, \bar{V}_P, \bar{P}_M, (E, A))$ is regular.

Notice that because $\sim$ is a congruence relation, if $A_i \in V_M \cup V_P$, $E_i \in \mathscr{E}$, and $E_i \subseteq A_i$ for $1 \leqslant i \leqslant n$, then there is a unique $E \in \mathscr{E}$ such that for any metaassignment h with $h(A_i) \in E_i$ for $1 \leqslant i \leqslant n$, $h(A_1 \cdots A_n) = h(A_1) \cdots h(A_n) \subseteq E_1 \cdots E_n \subseteq E$. This is the key observation that makes our construction work.

A rule $\langle A \rangle \to y$ is *lossless* if A appears in y and otherwise is *lossy*. Rules $Z \to y$ for Z a protovariable are always considered lossless. If $\langle A \rangle \to uAv$ is a hyperrule with $A \in V_M$ and $u, v \in (V_P \cup V_M \cup \{\langle \beta \rangle \mid \beta \in (V_P \cup V_M)^+)^*$, then the rule is said to "deposit" A. If $\langle \beta \rangle$ appears in a hyperrule, we must guess whether we will apply a series of lossless rules to $\langle \beta \rangle$ followed by a depositing lossless rule or whether we will apply zero or more nondepositing lossless rules followed by a lossy rule. In the first case we merely replace each $A \in V_M$ by some symbol (E, A) for $E \in \mathscr{E}_A$. In the second case we replace $\langle \beta \rangle$ by some $\hat{E}$ such that $L_\beta \cap E \neq \varnothing$.

More carefully, we construct the set of hyperrules $\bar{P}_h$ as follows. Consider any function f from V_M to $\mathscr{E}$ such that $f(A) \in \mathscr{E}_A$ for each $A \in V_M$; thus f assigns to each metavariable A an equivalence class contained in $\mathscr{E}_A$. Associated with f, define a homomorphism f_1 from $(V_M \cup V_P)^*$ into $(\bar{V}_M \cup \bar{V}_P)^*$ and a function f_2 from $(V_M \cup V_P)^*$ into $\mathscr{E}$ defined by $f_1(A) = (f(A), A)$ for $A \in V_M$ and $f_1(Z) = Z$ for $Z \in V_P$, $f_2(A) = f(A)$ for $A \in V_M$, $f_2(Z) = [Z]$ for $Z \in V_P$, and $f_2(xy) = f_2(x) \cdot f_2(y)$ for $x, y \in (V_M \cup V_P)^+$.

For each such function f and each hyperrule of P_h we add to $\bar{P}_h$ one or more hyperrules defined as follows. Consider a hyperrule in P_h:

$$\gamma \to u_0\langle v_1 \rangle \cdots u_{n-1}\langle v_n \rangle u_n$$

for $n \geqslant 0$, $u_i \in (V_P \cup V_M)^*$, $0 \leqslant i \leqslant n$, $v_i \in (V_P \cup V_M)^+$, $1 \leqslant i \leqslant n$, and $\gamma \in (V_P - \Sigma) \cup \{\langle A \rangle \mid A \in V_M\}$; if $n = 0$, the rule is $\gamma \to u_0$ and $u_0 \in (V_P \cup V_M)^+$. For $1 \leqslant i \leqslant n$ let $E_i = f_2(v_i)$.

Then $\bar{P}_h$ will contain all possible rules $\gamma' \to f_1(u_0)\, \bar{v}_1 \cdots f_1(u_{n-1})\, \bar{v}_n f_1(u_n)$ satisfying the conditions:

1. For $1 \leqslant i \leqslant n$, either $\bar{v}_i = \langle f_1(v_i) \rangle$ or $\bar{v}_i = \hat{E}_i$.
2. If $\gamma = Z \in V_P$, then $\gamma' = Z$.
3. If $\gamma = \langle A \rangle$, $A \in V_M$, $E = f(A)$, either (a) $\gamma' = \hat{E}$, and (E, A) does not appear in any $f_1(u_i)$ or any $\bar{v}_i$; or (b) $\gamma' = \langle (E, A) \rangle$, and (E, A) appears either in some $f_1(u_i)$ or some $\bar{v}_i$.

Let $G = (\bar{V}_M, \bar{V}_P, \Sigma, \bar{P}_M, \bar{P}_h, \sigma)$. Now condition 3(b) ensures that $\bar{G}$ is lossless. For E in $\mathscr{E}_A$, $L_{(E,A)} = L_{\hat{E}} = E - \{e\}$, so $L_{(E_1,A)} = L_{(E_2,B)}$ if and only if $E_1 = E_2$; otherwise, $L_{(E_1,A)} \cap L_{(E_2,B)} = \varnothing$. Suppose $\langle Y_1 \cdots Y_n \rangle$ appears in $\bar{P}_h$, each Y_i in $\bar{V}_P \cup \bar{V}_M$. If $Y_i \in V_P \subseteq \bar{V}_P$, let $E_i = [Y_i]$. Otherwise we must have $Y_i = (E_i, A_i)$ for some $A_i \in V_M$, $E_i \in \mathscr{E}_{A_i}$. So $L_{Y_1 \cdots Y_n} \subseteq L_{Y_1 \cdots Y_n} \subseteq E_1 \cdots E_n$ and there is a unique $E \in \mathscr{E}$ with $E_1 \cdots E_n \subseteq E$; thus $L_{Y_1 \cdots Y_n} \subseteq L_{\hat{E}}$. So $\bar{G}$ is factored; it is obviously normal and regular-based.

We have altered the hyperrules of G in two ways. We may replace a metavariable A of G by any metavariable (E, A) with $E \in \mathscr{E}_A$. Since $L_A = \bigcup_{E \in \mathscr{E}_A} E = \bigcup_{E \in \mathscr{E}_A} L_{(E,A)}$, this causes no problems. Notice that since $(E, A) \neq (E, B)$ for $A \neq B$, if A and B both appear in a hyperrule, no undesired duplications occur. Also, we might have functions f and f_2 as described above such that $\langle \beta \rangle$ is replaced by $\hat{f}_2(\beta) = \hat{E}$. In an actual production of G, $\langle \beta \rangle$ would appear as $\langle w \rangle$ for $w \in L_\beta \cap L_E$. Rule $\langle w \rangle \to y$ applied to $\langle w \rangle$ would either come from a lossy hyperrule $\langle A \rangle \to y'$ or from a lossless hyperrule $\langle A \rangle \to x \langle uAv \rangle z$ in which A was not deposited. In the first case all one needs to know is that $w \in L_A$—and recall that either $w \in L_A$ for *all* w in E or else $w \notin L_A$ for *all* w in E. In the second case the actual production looks like $\langle w \rangle \to x' \langle u'wv' \rangle y'$ and one needs only to know whether $u'wv' \in L_B$ for various B in V_M. Again, either $u'Ev' \subseteq L_B$ or $u'Ev' \cap L_B = \varnothing$, so it suffices to know that w is in equivalence class E.

Arguing along these lines, one can show that $L(G) = L(\bar{G})$. The actual proof is omitted since it is long and unenlightening. It involves showing by induction on the length of a derivation that for $w \in \Sigma^+$, $Z \in V_P - \Sigma$, and $y \in V_P{}^+$, $Z \Rightarrow_{\bar{G}}^* w$ if and only if $Z \Rightarrow_G^* w$ and $\langle y \rangle \Rightarrow_G^* w$ if and only if either $\langle y \rangle \Rightarrow_{\bar{G}}^* w$ or $\hat{E} \Rightarrow_{\bar{G}} w$, where $E = [y]$.

Now Baker's result immediately shows the following.

Corollary 1. If G is a normal, regular-based grammar, then $L(G)$ is context-sensitive.

Corollary 2. If G is a normal, regular-based grammar, then membership in $L(G)$ is decidable.

We can extend both corollaries. First let us show that some questions undecidable for context-sensitive grammars are decidable for normal, regular-based W-grammars.

Here are some concepts we shall find useful in the next few theorems.

Definition 3.4. Let $G = (V_M, V_P, \Sigma, P_M, P_h, \sigma)$ be a normal W-grammar and let $A \in V_M$. If $\gamma \to uAv \in P_h$, and $\gamma \neq \langle A \rangle$, then A *appears independently* in the hyperrule and the hyperrule is *creative*. For $w \in V_P{}^+$, let $L\langle w \rangle = \{ y \in \Sigma^+ \mid \langle w \rangle \Rightarrow^* y \}$ and $L\langle A \rangle = \bigcup_{w \in L_A} L\langle w \rangle$. For $\beta \in (V_M \cup V_P)^+$ let $L\langle \beta \rangle = \bigcup_{w \in L_\beta} L\langle w \rangle$. For $Z \in V_P - \Sigma$ let $L(Z) = \{ y \in \Sigma^+ \mid Z \Rightarrow^* y \}$.

We define reduced normal factored W-grammars. Our definition is in the same spirit as the usual definition of reduced context-free grammars.

Definition 3.5. A normal factored W-grammar $G = (V_M, V_P, \Sigma, P_M, P_h, \sigma)$ is *reduced* if either $V_M = \Sigma = P_M = P_h = \varnothing$ or:

1. For each $Z \in V_P - \Sigma$, $L(Z) \neq \varnothing$.
2. If $\langle \beta \rangle$ appears in a hyperrule, then for each $w \in L_\beta$, $L\langle w \rangle \neq \varnothing$.
3. For each $Z \in V_P$ there are words $u, v \in \Sigma^*$ such that $\sigma \Rightarrow^* uZv$.
4. If $\langle A \rangle$ appears on the left-hand side of a hyperrule, there are $w \in L_A$, $u, v \in \Sigma^*$, such that $\sigma \Rightarrow^* u\langle w \rangle v$.
5. For $A \in V_M$, $L_A \subseteq (V_P - \{\sigma\})^+$, $L_A \cap (V_P - \Sigma) = \varnothing$ and L_A is infinite.
6. The start symbol σ does not appear on the right-hand side of a hyperrule.
7. There are no hyperrules $\langle A \rangle \to \langle A \rangle$ or $\langle Z \rangle \to \langle Y \rangle$.

Theorem 3.2. Given a normal regular-based W-grammar G, we can construct a reduced, lossless, factored, normal, regular-based W-grammar $\bar{G}$ such that $L(G) = L(\bar{G})$ and $\bar{G}$ is regular or strict or unary if G is.

Proof. We can of course assume $G = (V_M, V_P, \Sigma, P_M, P_h, \sigma)$ to be lossless, factored, normal, and regular-based.

First we can ensure that for $A \in V_M$, L_A contains no protovariable Z. For if $Z \in L_A$, we can first replace L_A by $L_A - \{Z\}$ and G will still be regular-based and factored. Then we add a new symbol $\hat{Z}$ to $V_P - \Sigma$ and for each hyperrule $\gamma \to y$ add the result of first replacing A by Z in $\gamma \to y$ and then replacing $\langle Z \rangle$ by $\hat{Z}$. So we can assume that $L_A \cap (V_P - \Sigma) = \varnothing$. By the standard methods for context-free grammars, we can ensure that $L_A \subseteq (V_P - \{\sigma\})^+$ and (6) holds.

Let

$$V = \{ Z \in V_P - \Sigma \mid L(Z) \neq \varnothing \}$$

Ideally, we should like to have $V_P - \Sigma = V$, since the appearance of a symbol of $V_P - \Sigma - V$ obviously blocks a derivation in a lossless grammar. First we must locate V.

Notice that because G is factored, if $w \in L_A$ and $L\langle w \rangle \neq \varnothing$, then

$L\langle w'\rangle \neq \varnothing$ for any $w' \in L_A \cap (\Sigma \cup V)^+$. Let $T = \{A \in V_M \mid L\langle A\rangle \neq \varnothing\}$, and $I = \{A \in V_M \mid L_A \cap (\Sigma \cup V)^+ \neq \varnothing\}$.

We construct V, T, and I in a familiar way. First, let $V_1 = T_1 = \varnothing$ and $I_1 = \{A \in V_M \mid L_A \cap \Sigma^+ \neq \varnothing\}$. If we have built V_n, T_n, and I_n with $V_n \subseteq V$, $T_n \subseteq T$, and $I_n \subseteq I$, we obtain V_{n+1}, T_{n+1}, and I_{n+1} as follows, starting with $V_n \subseteq V_{n+1}$, and $T_n \subseteq T_{n+1}$. We search P_n for a rule

$$\gamma \to u_0\langle v_1\rangle \cdots u_{m-1}\langle v_m\rangle u_m$$

with u_i, v_j, as usual, satisfying:

(a) $u_0, u_1, \ldots, u_m \in (V_n \cup I_n \cup \Sigma)^+$.

(b) $v_1, \ldots, v_m \in (V_n \cup I_n \cup \Sigma)^+$.

(c) For $1 \leqslant j \leqslant n$, $L_{v_j} \subseteq L_{A_j}$ for some $A_j \in T_n$.

Then if $\gamma = Z \in V_P$, we add Z to V_{n+1}; if $\gamma = \langle A\rangle$, we add to T_{n+1}, along with any B with $L_B = L_A$. When this is done let

$$I_{n+1} = \{A \in V_M \mid L_A \cap (V_{n+1} \cup \Sigma)^+ \neq \varnothing\}$$

By construction we have $V_n \subseteq V_{n+1} \subseteq V$, $T_n \subseteq T_{n+1} \subseteq T$, and $I_n \subseteq I_{n+1} \subseteq I$.

Since $\mid V_P \cup V_M \mid$ is finite, there is an $n_0 \leqslant \mid V_P \cup V_M \mid$ such that $V_{n_0} \cup T_{n_0} = V_{n_0+1} \cup T_{n_0+1}$. Then $V_{n_0} = V_{n_0+1}$, $T_{n_0} = T_{n_0+1}$, and $I_{n_0} = I_{n_0+1}$. Hence $V_{n_0} = \bigcup_n V_n \subseteq V$, $T_{n_0} = \bigcup_n T_n \subseteq T$, and $I_{n_0} = \bigcup_n I_n \subseteq I$. We omit the straightforward but long proof that $V_{n_0} = V$, $T_{n_0} = T$, and $I_{n_0} = I$.

If $\sigma \notin V$, then $L(\sigma) = L(G) \neq \varnothing$. In this case let $\bar{G}$ be the trivial W-grammar $\bar{G} = (\{\sigma\}, \varnothing, \varnothing, \varnothing, \varnothing, \sigma)$. Suppose $\sigma \in V$.

Now we can alter G to ensure that $V_P \subseteq V \cup \Sigma$ and $\varnothing \neq L_A \subseteq (V \cup \Sigma)^+$ for each $A \in V_M$; the construction is quite obvious; when we eliminate a symbol we obviously eliminate all members of $P_M \cup P_h$ in which it appears. We also eliminate any hyperrule containing a hypernotion $\langle\beta\rangle$ with $L_\beta \subseteq L_A$ for some A in $V_M - (I \cup T)$. Then we can assume that G satisfied 1 and 2.

Now we want to ensure that 3 and 4 hold. We want to construct sets similar to I, T, and V; this time we only sketch the construction. We want to build sets J (consisting of metavariables appearing independently in hyperrules), K (consisting of metavariables satisfying 3), and N (consisting of protovariables and terminals satisfying 3).

We start with $J = K = \varnothing$ and $N = \{\sigma\}$. We alternate scanning hyperrules and sets L_A. If $Z \to y$ is a hyperrule and $Z \in N$, we add to J all matavariables in y, to N all members of V_P in y, and to K any A such that y contains $\langle\beta\rangle$ and $L_\beta \subseteq L_A$. If $\langle A\rangle \to y$ is a hyperrule, we add to N and K as above and add to J any symbol in $V_M - \{A\}$ appearing in y. When we add A to J, we add to N any $Z \in V_P$ with $L_A \cap V^*ZV^* \neq \varnothing$; since the L_A are context-free (in fact regular), this condition is testable. Eventually the process ends when we scan all hyperrules without increasing J, K, or N.

When we finish we simply eliminate all members of $V_P - N$ everywhere in G and all hyperrules $\langle A\rangle \to y$ where $A \notin K$. When we have finished G satisfies 3 and 4 still satisfies 1 and 2.

Next, if $L_A = \{w_1, \ldots, w_n\}$ and $A \in V_M$, replace each hyperrule $\gamma \to y$ by n hyperrules

$$h_i(\gamma) \to h_i(y)$$

where h_i is a homomorphism such that $h_i(A) = w_i$ and h_i is the identity elsewhere. Do this successively for each $A \in V_M$ with L_A finite. This may create an "illegal" hyperrule

$$\langle w\rangle \to y$$

for $w \in V_P{}^+$. In this case eliminate this rule, create a new protovariable $\hat{w}$, add a rule $\hat{w} \to y$, and replace every occurrence of $\langle w\rangle$ by $\hat{w}$. Hence we can assume that L_A is infinite. Thus, using the remarks at the start of the construction, we see that 5 and 6 hold.

Finally, we can satisfy 7 with constructions similar to those used for context-free grammars.[15] Thus we can construct a new W-grammar $\bar{G}$ which is reduced and still lossless, factored, normal, and regular-based and which also generates $L(G)$. $\square$

Now we state our main decidability results.

Theorem 3.3. It is decidable for normal regular-based grammars G whether $L(G)$ is empty and whether $L(G)$ is finite.

Proof. We can assume that $G = (V_M, V_P, \Sigma, P_M, P_h, \sigma)$ is reduced, factored, and lossless as well as normal and regular-based. Obviously $L(G) \neq \varnothing$ if and only if G is not the trivial grammar $(\varnothing, \{\sigma\}, \varnothing, \varnothing, \varnothing, \sigma)$.

Assume $L(G) \neq \varnothing$. First notice that if $L\langle A\rangle \neq \varnothing$ and L_A is infinite, then $L\langle A\rangle$ is infinite. For if $x \in L\langle A\rangle$, then there is a word $w \in L_A$ such that $|w| > |x|$. As we mentioned before, the fact that G is factored implies that since $L\langle A\rangle \neq \varnothing$, $L\langle w\rangle$ is also nonempty. But G is lossless, so if $y \in L\langle w\rangle$, then $y \in L\langle A\rangle$ and $|y| \geqslant |w| > |x|$. Hence $L\langle A\rangle$ is infinite.

Since G is reduced, L_A is infinite for $A \in V_M$. Further, every hyperrule in G is "usable" in some completed derivation. Hence if G has any creative hyperrule, $L(G)$ is infinite.

So assume now that no hyperrule is creative—no metavariables ever appear independently. This means that there are at any point only finitely many actual productions applicable to a string generated from σ. Hence for any q there are only finitely many complete derivation trees in which no path has more than q nodes. Call a complete derivation tree *minimal* if there is no smaller tree yielding the same word in $L(G)$. By building all smaller derivation trees, one can certainly determine whether a given tree is minimal.

Suppose G has a minimal complete derivation tree with a path containing

$$q = 2 + | V_P - \Sigma | + | V_M | + (| V_P - \Sigma |)(| V_M |)$$

nodes. If the path contains $1 + | V_P - \Sigma |$ nodes labeled in $V_P - \Sigma$, two of them must have the same label, say Z. Thus there is a subderivation $Z \Rightarrow^* uZv$, $u, v \in \Sigma^*$. If $u = v = e$, the tree is not minimal. If $uv \neq e$, then we have $\sigma \Rightarrow^* w_1 Z w_2$ and $Z \Rightarrow^* w_3$ for w_1, w_2, $w_3 \in \Sigma^*$ and so $L(G)$ has the infinite context-free subset $\{w_1 u^n w_3 v^n w_2 \mid n \geqslant 0\}$.

Otherwise, the path must contain two nodes labeled $\langle w_1 \rangle$ and $\langle w_2 \rangle$ such that w_1, $w_2 \in L_A$ for some $A \in V_M$ and no intermediate node has a label in $V_P - \Sigma$. We could apply to $\langle w_2 \rangle$ a production derived from the same hyperrules that one applied to $\langle w_1 \rangle$ and continue in this way until we generate some $\langle w_3 \rangle$ with w_3 in L_A. Repeating the process as long as we choose, we see that $L(G)$ is infinite.

On the other hand, if no complete derivation tree has a path with q or more nodes, then $L(G)$ is finite. Hence we can tell whether $L(G)$ is finite. □

4. RELATIONS BETWEEN W_{RB} AND OTHER FAMILIES

In this section we show W_{RB} to be contained in two familiar families of languages.

First we show that W_{RB} is a subfamily of the family of indexed languages —the languages defined by indexed grammars,[1] by nested stack automata,[2] and by OI (outside-in) macrogrammars.[11,12] It appears simpler and more enlightening to use macrogrammars.

We shall discuss macrogrammars briefly and informally. A very careful and rigorous treatment appears in Ref. 11.

A *macrogrammar* contains three disjoint sets of symbols—a finite vocabulary Σ of *terminals*, a finite vocabulary I of function letters ranked by a function ρ, and a finite set V of *variables*. Terms are defined inductively. Any member of $\Sigma \cup V$ is a *term*, as is any zero-place function letter A [i.e., $A \in I$ and $\rho(A) = 0$]. If α and β are terms, so is $\alpha\beta$. If $\alpha_1, ..., \alpha_n$ are terms, $n \geqslant 1$, $F \in I$, and $\rho(F) = n$, then $F(\alpha_1, ..., \alpha_n)$ is a term. A macrogrammar consists of Σ, I, ρ, and V plus a start symbol σ usually taken to be 0-place [i.e., $\sigma \in I$ and $\rho(\sigma) = 0$] and a finite set P of productions of the forms $Z \rightarrow y_1$ and $F(x_1, ..., x_n) \rightarrow y_2$, where F, $Z \in I$, $\rho(Z) = 0$, $\rho(F) = n$, the x_i are all distinct members of V (so $x_i \neq x_j$ for $i \neq j$), and y_1 and y_2 are terms such that $y_1 \in (I \cup \Sigma)^+$ and $y_2 \in (I \cup \Sigma \cup \{x_1, ..., x_n\})^+$.

Derivations in a macrogrammar could go from the outside in or the inside out, with different results. We shall only consider outside-in, OI, derivations since those yield precisely the indexed languages. If $F(x_1, ..., x_n) \to y$ is a production, $\alpha_1, ..., \alpha_n$ are terms, and y' is obtained from y by substituting α_i for x_i [i.e., $y' = h(y)$, where $h(x_i) = \alpha_i$, $1 \leqslant i \leqslant n$, and h is the identity elsewhere], then $F(\alpha_1, ..., \alpha_n) \Rightarrow y'$. If $Z \to y$ is a rule, $\rho(Z) = 0$, then $Z \Rightarrow y$. If $u \Rightarrow v$ and x and y are terms, then $xuy \Rightarrow xvy$. We extend $\Rightarrow$ to $\Rightarrow^*$ in the usual way. Then the *language generated by* G is $L(G) = \{w \in \Sigma^+ \mid \sigma \Rightarrow^* w\}$.

For example, the set $\{a^{n^2} \mid n \geqslant 1\}$ is generated by the macroproductions $\sigma \to F(a, a)$, $F(x_1, x_2) \to F(x_1x_2aa, x_2aa)$, and $F(x_1, x_2) \to x_1$. The string a^9 is obtained:

$$\sigma \Rightarrow F(a, a) \Rightarrow F(a^4, a^3) \Rightarrow F(a^9, a^5) \Rightarrow a^9$$

Theorem 4.1. The family W_{RB} is contained in the family of indexed languages. Given a normal regular-based grammar G, we can construct a OI macrogrammar $\bar{G}$ such that $L(G) = L(\bar{G})$.

Proof. We can assume that $G = (V_M, V_P, \Sigma, P_M, P_h, \sigma)$ is factored. Let $V_M = \{A_1, ..., A_n\}$. Each metavariable A_i will become an n-place function letter in $\bar{G}$ and each protovariable Z will be represented in $\bar{G}$ by a zero-place letter Z and an n-place letter $\hat{Z}$. We shall generate in $\bar{G}$ terms such as $A_i(\alpha_1, ..., \alpha_n)$ where each α_j is in L_{A_j} and G is currently applying a production to $\langle\alpha_i\rangle$. The idea is that a hyperrule such as $\langle A_1\rangle \to aA_2\langle A_1aA_2A_1\rangle$ with $L_{A_1aA_2A_1} \subseteq L_{A_3}$ would become a macroproduction

$$A_1(x_1, x_2, x_3) \to ax_2A_3(y_1, y_2, x_1ax_2x_1)$$

and the y_i would generate members of L_{A_i}. This will not quite work—we need outside-in derivation to make proper duplicates but in order to expand y_1 *first*, we would have to use inside-out derivation. So our construction must become more complicated.

Let T be the set of all vectors $(B_1, ..., B_n)$ with each B_i in V_M. Let $V = \{x_1, ..., x_n\}$ and

$$\begin{aligned} I = \{A_i, A_{i,t} \mid 1 \leqslant i \leqslant n, t \in T\} \cup \{S_0\} \\ \cup \{Z, \hat{Z}, \hat{Z}_t \mid Z \in V_P - \Sigma, t \in T\} \cup \{S, S_t \mid t \in T\} \end{aligned}$$

with $\rho(A_i) = \rho(A_{i,t}) = \rho(\hat{Z}) = \rho(\hat{Z}_t) = n$ and $\rho(Z) = 0$ for $A_i \in V_M$, $Z \in V_P - \Sigma$, $t \in T$. Now a term $A_{i,t}(\alpha_1, ..., \alpha_t)$ with $t = (B_1, ..., B_n)$ but $B_i = A_i$ will mean that we want to expand $\langle\alpha_i\rangle$ next, α_j is in L_{B_j} for $i \neq j$, and we are trying to expand α_j to a member of L_{A_j}.

The macroproductions that simulate P_M are relatively easy to state. If

$1 \leqslant k \leqslant n$, $t = (B_1, ..., B_n) \in T$, $B_k = A_k$, but B_i is arbitrary for $i \neq k$, then we need all possible productions

$$A_{k,t}(x_1, ..., x_n) \to A_{k,t'}(y_1, ..., y_n)$$

where $t' = (C_1, ..., C_n)$, $C_k = B_k = A_k$, $y_k = x_k$, and for $i \neq k$ either $C_i = B_i$ and $y_i = x_i$ or else $y_i = wx_i$ and $C_i \to wB_i$ is a metaproduction in P_M. We also need for $t_0 = (A_1, A_2, ..., A_n)$ the production

$$A_{k,t_0}(x_1, ..., x_n) \to A_k(x_1, ..., x_n)$$

Similarly, for $Z \in V_P - \Sigma$ and $t = (B_1, ..., B_n)$ we need all productions

1. $Z \to \hat{Z}_t(w_1, ..., w_n)$ where $B_j \to w_j$, $w_j \in V_P{}^+$ for $1 \leqslant j \leqslant n$.
2. $\hat{Z}_t(x_1, ..., x_n) \to \hat{Z}_{t'}(y_1, ..., y_n)$ where $t' = (C_1, ..., C_n)$ and for $1 \leqslant j \leqslant n$ either $C_j = B_j$ and $y_j = x_j$ or else $y_j = w_jx_j$ and $C_j \to w_jB_j$ in P_M.

For $t_0 = (A_1, ..., A_n)$ we need $\hat{Z}_{t_0}(x_1, ..., x_n) \to \hat{Z}(x_1, ..., x_n)$.

Each hyperrule in P_h yields a number of macroproductions. Instead of giving the general form, which would bristle with subscripts, let us give two examples, for $n = 3$. Suppose P_h contains

$$\langle A_2 \rangle \to A_2a\langle aA_1A_2\rangle\, b\langle A_3A_2a\rangle A_1$$

and $L_{aA_1A_2} \subseteq L_{A_1}$ and $L_{A_3A_2a} \subseteq L_{A_2}$. Then we have all possible macroproductions

$$A_2(x_1, x_2, x_3) \to x_2aA_{1,t}(ax_1x_2, u_2, u_3)\, bA_{2,r}(v_1, x_3x_2a, v_3)x_1$$

where $t = (A_1, B_2, B_3)$; $r = (C_1, A_2, C_3)$; $v_1, v_3, u_2, u_3 \in V_P{}^+$; and $B_2 \to u_2$, $B_3 \to u_3$, $C_1 \to v_1$, and $C_3 \to v_3$ are in P_M. Similarly, if $Z \in V_P$ and we have a hyperrule

$$Z \to aA_2\langle bA_1A_1b\rangle\, A_3Z$$

with $L_{cA_1A_1b} \subseteq L_{A_3}$, we have all possible macroproductions

$$\hat{Z}(x_1, x_2, x_3) \to ax_2A_{3,t}(u_1, u_2, bx_1x_1b)\, x_3Z$$

with $t = (B_1, B_2, A_3)$; $u_1, u_2 \in \Sigma^+$; and $B_1 \to u_1$, $B_2 \to u_2$ in P_M.

The reader can verify that the OI macrogrammar so constructed generates $L(G)$. $\square$

Corollary. W_{RB} is properly contained in the family of context-sensitive languages.

A language L in W_{RB} is, as we saw, context-sensitive and so can be recognized by a nondeterministic multitape Turing machine which needs at most $|w|$ tape squares to accept w. In fact, L can not only be recognized, nondeterministically, in linear space but also in linear time. For W_{RB} is contained in $\mathcal{Q}$, the family of languages accepted in *realtime* by nondeterministic multitape Turing machines. Languages in $\mathcal{Q}$ are called *quasirealtime.* More rigorous definitions of multitape Turing machines, nondeterminism, realtime, and quasirealtime can be found in Ref. 8.

It seems plausible that containment in $\mathcal{Q}$ should be a consequence of the last theorem. We conjecture that the indexed languages are indeed quasirealtime. However, to the best of the author's knowledge, this has never been established in print and must be considered to be an open question. Thus $W_{RB} \subseteq \mathcal{Q}$ requires an independent proof.

We shall not give the full proof that $W_{RB} \subseteq \mathcal{Q}$, since it is very long and would lead us far afield of our goals. Instead we content ourselves with sketching some of the ideas that make the proof "work."

Theorem 4.2. $W_{RB} \subseteq \mathcal{Q}$; given a normal regular-based grammar G, one can construct a nondeterministic multitape Turing machine to accept $L(G)$ in realtime.

Proof. We only outline the necessary construction.

First, the results in Refs. 6–8 indicate that to show L in $\mathcal{Q}$ it suffices to exhibit a nondeterministic multitape Turing machine M and an integer k such that M generates L and M generates w in time $k\,|\,w\,|$. That is, linear time is no more powerful than realtime for *nondeterministic* Turing machines. The results of Refs. 8 and 18 show that, without loss of generality, we can let M have any finite number of working tapes and any finite number of read–write heads per tape. Thus, using auxiliary two-headed tapes, we can copy and duplicate without loss of time.

We can assume $L = L(G)$ for a reduced, factored, lossless, normal, regular-based W-grammar $G = (V_M, V_P, \Sigma, P_M, P_h, \sigma)$. The idea is that in any cycle M selects a hyperrule $\gamma \to y$, a metaassignment h, and a production $h(\gamma) \to h(y)$ and applies this to $h(\gamma)$. What is involved in obtaining $h(y)$ from y? Since each G_A is regular, one can, nondeterministically, select $h(A) \in L_A$ in the time it takes to write $|\,h(A)|$, thus checking that $h(A) \in L_A$ causes no loss of time. If A appears twice in y, then M must duplicate $h(A)$. By using an auxiliary two-headed tape, M can write $h(A)\,h(A)$ in time $2\,|\,h(A)|$. It will need at most $|\,V_M\,|$ such tapes; at the end of each cycle M can erase these tapes at the cost of doubling computation time.

The organization of this simulation is of some importance. There are three main tapes, TL for the left part of the generated string, TR for the right part, and TC for the current section to be expanded. If a protovariable Z is being expanded, there is no problem selecting which hyperrules are legal. If $\langle w \rangle$ is to be expanded, the contents of TC are written as $\langle_A w_A \rangle$ where $w \in L_A$; by suitably recoding one could write this in $|w|$ tape squares. Thus M can select a hyperrule $\langle B \rangle \rightarrow y$ for $L_A = L_B$ without testing whether $w \in L_B$. Since G is factored, if $\langle \beta \rangle$ appears in the right-hand side of a hyperrule, M knows at once the set $V_\beta = \{A \in V_M \mid L_\beta \subseteq L_A\}$ and knows that if $A \notin V_\beta$, then $h(\beta) \notin L_A$ for any metaassignment h. So M can select at random $A \in V_\beta$ and write $\langle_A h(\beta)_A \rangle$ *without* inspecting $h(\beta)$. This is a vital point. The other point is that since G is lossless, when $\langle w \rangle$ is expanded w remains behind in some form or other.

The order of expansion is of some significance. When the production applied is $Z \rightarrow h(y)$ for $Z \in V_P - \Sigma$ and $h(y) \notin \Sigma^+$, it does not matter which part of $h(y)$ is expanded next, so select the leftmost. That is, if $y = u\gamma v$, $u \in \Sigma^+$, and $\gamma \in (V_P - \Sigma) \cup \{\langle w \rangle \mid w \in V_P{}^+\}$, put u on TL, v on TR, and the appropriate encoding of $\langle w \rangle$ (i.e., $\langle_A w_A \rangle$ for $w \in L_A$) on TC. If the hyperrule is $\langle A \rangle \rightarrow y$, there is a slight catch. If $y = x\langle uAv \rangle Z$ and TC contains $\langle_A w_A \rangle$ (or $\langle_C w_C \rangle$ for $L_A = L_C$), write $h(x)$ on TL, $h(z)$ on TR, and $\langle_B h(u)\, w h(v)_B \rangle$ on TC for $L_{uAv} \subseteq L_B$. Now the latter operation need take only $|h(u)| + |h(v)|$ steps because TC can also be a two-headed tape, and because M does not have to scan $h(u)\, w h(v)$ to know that it is in L_B.

One other point should be mentioned. If the hyperrule $\gamma \rightarrow y$ yields a production with $h(y) \in \Sigma^+$, M must guess whether or not TL has anything left to expand. A wrong guess will cause M to block. If M guesses that TL has nothing to expand, it puts $h(y)$ on the right of TL and it transfers terminal symbols from the left of TR to the right of TL until it hits $\gamma = Z \in V_P - \Sigma$ or $\gamma = \langle_A w_A \rangle$. Then it puts a barrier $\square$ on TL and puts γ on TC. On the other hand, if M guesses TL contains a suitable γ for expansion, it puts $h(y)$ on TR and transfers terminals from the right of TL to the left of TR until either it empties TL or it encounters the barrier $\square$ or it finds γ. In the first two cases M blocks. In the last case M puts γ on TC and continues. This avoids repeated shuttling of symbols between TL and TR—once a symbol goes from TR to TL it can never be moved until the final cleanup.

Finally, the process starts with σ on TC and TL and TR empty and ends when M guesses that only terminals remain. A wrong guess means a block. In the final cleanup M transfers TL and TR minus barrier $\square$ onto an output tape. Since there can never be more barriers than terminals and since TL and TR could have two heads, this takes at most $2|w|$ steps for an output w and so at worst increases the linear factor by one. The way to see that the process is linear is to count not the steps per cycle but the number of times a symbol is "handled" (created, transferred from TR to TC, etc.), recalling that G is lossless; in the worst case no symbol is handled more than eight times before yielding a new terminal on the output tape. □

Corollary 1. Each member of W_{RB} can be expressed as the nonerasing homomorphic image of the intersection of three context-free languages.

Proof. This is true of $\mathscr{Q}$.[8] □

Corollary 2. W_{RB} is contained in the family of deterministic context-sensitive languages.

Proof. This follows from Corollary 1, since the family of deterministic context-sensitive languages is closed under intersection and nonerasing homomorphism. □

5. NEGATIVE RESULTS FOR W_{RB}

We have examined what normal regular-based grammars *can* do, what useful normal forms and reductions apply to them, and how easy they are to generate or recongize. Now we discuss their limitations. First we exhibit a language in $W_{RB} - W_R$ and then notice that the same arguments can exhibit a language in $\mathscr{Q} - W_{RB}$. The languages are those of Examples 4 and 5. Then we discuss briefly the closure properties of W_{RB}.

Theorem 5.1. The language $L = \{a_1{}^n a_2{}^n a_3{}^n a_4{}^n \mid n \geqslant 1\}$ is not in W_R.

Proof. The proof turns on some implications of the structure of L. First notice that if L contains a subset $\{u_1 v_1{}^n u_2 v_2{}^n u_3 v_3{}^n u_4 \mid n \geqslant 1\}$, then we must have $v_1 = v_2 = v_3 = e$ for we cannot alter the occurrences of three symbols and leave the fourth unchanged. A fortiori, L cannot contain an infinite context-free or an infinite regular subset.[5]

Part of a word in L may uniquely determine the rest. Given a word u containing a_2, there is at most one v such that uv is in L. Hence if uv_1 and uv_2 are in L, $v_1 = v_2$; similarly, $u_1 v$ and $u_2 v$ in L implies $u_1 = u_2$ if v contains a_3. The situation regarding middle sections is slightly more complicated. If w contains three distinct letters (i.e., $w \in a_1{}^+ a_2{}^+ a_3{}^+ \cup a_2{}^+ a_3{}^+ a_4{}^+ \cup a_1{}^+ a_2{}^+ a_3{}^+ a_4{}^+$), then there is at most one u and one v with uwv in L. If u contains two letters, then $uwv \in L$ uniquely determines wv and similarly for v and uw. So if any three of $u_1 w_1 v_1$, $u_1 w_2 v_1$, and $u_2 w_2 v_2$ are in L, either $w_1 = w_2$ or $(u_1, v_1) = (u_2, v_2)$.

We also need the fact that although duplication does not preserve regularity in general, if R is regular and $R \subseteq w^*$ for any word w, then

$$\mathrm{Dup}(R, k) = \{y^k \mid y \in R\}$$

is regular for any $k \geqslant 1$. Also note that no subword with more than one letter can be duplicated in L, so if we use a hyperrule with two occurrences of a metavariable A, we may as well assume that $L_A \subseteq a_i^+$ for some i.

If L is in W_R, we can assume that $L = L(G)$ for a reduced, factored, lossless, normal, regular-based grammar $G = (V_M, V_P, \Sigma, P_M, P_h, \sigma)$ such that the rules of P_h are of the forms

$$\gamma \to w, \qquad \gamma \to u\langle w\rangle, \qquad \gamma \to uY$$

for $w \in (V_M \cup \Sigma)^+$, $u \in (V_M \cup \Sigma)^*$, and $\gamma \in V_P - \Sigma$ or $\gamma = \langle A\rangle$ for $A \in V_M$.

Observe that since G is lossless and factored, if $\langle w_1\rangle \Rightarrow^* z$, $z \in \Sigma^+$, and $w_1 \in L_A$, then z can be factored $z = z_1 w_1 z_2 w_1 \cdots z_k w_1 z_{k+1}$ in such a way that for any $w_2 \in L_A$, $\langle w_2\rangle \Rightarrow^* z_1 w_2 z_2 \cdots z_k w_2 z_{k+1}$ and $k = 1$ if no duplications occur. Also, if $\langle w_1\rangle \Rightarrow^* \langle u w_1 v\rangle$ without duplications and w_1 and $u w_1 v \in L_A$, then $\langle w_1\rangle \Rightarrow^* \langle u^n w_1 v^n\rangle$ for all $n \geqslant 1$.

First we show that σ is really the only useful protovariable in G. For $Z \in V_P - \Sigma$, $A \in V_M$, and $w \in L_A$, let $\pi(Z) = \{(u, v) \mid u, v \in \Sigma^+, \sigma \Rightarrow^* uZv\}$. Now protovariables behave like nonterminals in an ordinary context-free grammar, so $L(G)$ contains $uL(Z)v$ for each $u, v \in \pi(Z)$. Because G is reduced, $L(Z) \neq \varnothing \neq \pi(Z)$, either $\pi(Z) = \{(u_0, v_0)\}$ [i.e., $|\pi(Z)| = 1$] or $L(Z) = \{w_0\}$ [i.e., $|L(Z)| = 1$]. In the first case we can add $\{\sigma \to u_0 y v_0 \mid Z \to y \text{ is in } P_h\}$ and then remove Z from G. In the second case we can replace Z by w_0 on the right-hand side of hyperrules and eliminate all hyperrules containing Z; however, if Z appears in a word of L_A and A appears in some $\langle\beta\rangle$, we may have to add $Z \to w_0$ as the only hyperrule involving Z. In the latter case we regard Z as a specially tagged version of w_0 and so, in effect, in Σ^+. Thus in the subsequent argument we lose no generality by assuming $V_P = \Sigma \cup \{\sigma\}$.

Recall that a metavariable A appears independently in a hyperrule $\gamma \to y$ if A appears in y but not in γ. We now observe that we can assume that metavariables do not appear independently in hyperrules of G. Suppose $\sigma \to u_1 A \cdots u_k A u_{k+1}\langle\beta\rangle$ and A does not appear in $u_1, \ldots, u_{k+1}$ or β. As mentioned before, if $k \geqslant 2$, we can assume that $L_A \subseteq a_i^+$ for some i and rearrange the rule as $\sigma \to u_1 u_2 \cdots u_k A^k u_{k+1}\langle\beta\rangle$. For metaassignment h and $\langle h(\beta)\rangle \Rightarrow^* w$, $w \in \Sigma^+$, $h(u_1 \cdots u_k)\,\mathrm{Dup}(L_A, k)\,h(u_{k+1})w$ is an infinite regular subset of $L(G)$, a contradiction. Suppose $\sigma \to u\langle\beta\rangle$ is a hyperrule and u and β contain A. As before, we can assume $u = u_1 A^k$ and $\beta = A^r \beta_1$ where $k, r \geqslant 1$ and A does not appear in u_1, u_2, or β_1. If $\langle w_1{}^r h(\beta_1)\rangle \Rightarrow^* z \in \Sigma^+$ for $w_1 \in L_A$ and a metaassignment h, we can factor $z = z_1 w_1{}^r \cdots z_s w_1{}^r z_{s+1}$ such that

$$\langle w_2{}^r h(\beta_1)\rangle \overset{*}{\Rightarrow} z_1 w_2{}^r \cdots z_s w_2{}^r z_{s+1}$$

for any $w_2 \in L_A$, and $z_1 w_2{}^r \cdots z_s w_2{}^r z_{s+1} = w_2{}^r z_1 \cdots z_{s+1}$. Thus

$$h(u_1)\,\mathrm{Dup}(L_A, k + rs)\,z_1 \cdots z_{s+1}$$

is an infinite regular subset of G. By examining the case $\sigma \to u\langle\beta\rangle$ where A appears in β but not u and all the cases $\langle B\rangle \to y$ where $A \neq B$ and A appears in y, we find in all cases, if G contains a creative hyperrule, then $L(G)$ contains an infinite regular subset of the form $x\,\mathrm{Dup}(L_A, k)\,y$ where $L_A \subseteq a_i^+$ for $k \geqslant 2$. Hence we can assume that no metavariable appears independently in a hyperrule.

We can also eliminate rules $\langle A\rangle \to u\langle\beta\rangle$ where A does not appear in β. As noted, we can assume that $u \in (\Sigma \cup \{A\})^+$ and $\beta \in \Sigma^+$. Suppose there is a metaassignment h and a string $v \in \Sigma^+$ such that $\sigma \Rightarrow^* v\langle h(A)\rangle$. Then $vh(u)L\langle\beta\rangle \subseteq L(G)$. Either $|L\langle\beta\rangle| = 1$ or there are unique v and h with $\sigma \Rightarrow^* v\langle h(A)\rangle$. In the first case we replace $\langle A\rangle \to u\langle\beta\rangle$ by $\langle A\rangle \to uw_0$ for $L\langle\beta\rangle = \{w_0\}$ and in the second case we remove $\langle A\rangle \to u\langle\beta\rangle$ and add $\sigma \to vh(u)\langle\beta\rangle$.

Thus we can assume that G contains only hyperrules of the forms

$$\sigma \to w, \qquad \sigma \to u\langle w\rangle v, \qquad \langle A\rangle \to xAy, \qquad \langle A\rangle \to x\langle v_1Av_2\rangle$$

for $w \in \Sigma^+$; $u, v \in \Sigma^*$; $x, y, v_1, v_2 \in (\Sigma \cup \{A\})^*$. In the last case we can assume $v_1v_2 \neq e$, for $x = v_1 = v_2 = e$ would be trivial ($\langle A\rangle \to \langle A\rangle$) and $v_1 = v_2 = e$ but $x \neq e$ would, if used, produce an infinite regular subset of the form

$$w_1x^*w_2$$

Since $L(G) = L$ is infinite, there must be a $w \in \Sigma^+$ such that $L\langle w\rangle$ is infinite and $\sigma \to u\langle w\rangle v$ is a rule. Any derivation from $\langle w\rangle$ must look like

$$\langle w\rangle \Rightarrow u_1\langle w_1\rangle v_1 \Rightarrow \cdots \Rightarrow u_n\langle w_n\rangle v_n \Rightarrow x$$

for $u_i \in \Sigma^*$, w_i, $x \in \Sigma^+$, where each w_i is a proper substring of w_{i+1} and for each i we can factor x as $x = x_1w_i \cdots x_rw_ix_{r+1}$ where if w_i, v are in L_A, then $\langle v\rangle \Rightarrow^* x_1v \cdots x_rvx_{r+1}$. Since there are no independent metavariables in P_h, there are only finitely many derivations of size n from $\langle w\rangle$. Since V_M is finite, if $L\langle w\rangle$ is infinite, we must have

$$\langle w\rangle \overset{*}{\Rightarrow} u_1\langle w_1\rangle z \overset{*}{\Rightarrow} u_1u_2\langle w_2\rangle z \overset{*}{\Rightarrow} u_1u_2xz \tag{1}$$

for $w_1, w_2, x \in \Sigma^+$; $u_1, u_2 \in \Sigma^*$; and w_1 and w_2 in L_A for the same metavariable A. Then $\sigma \Rightarrow^* uu_1\langle w_1\rangle z$, $\langle w_1\rangle \Rightarrow^* u_2\langle w_2\rangle$, and $\langle w_2\rangle \Rightarrow^* x$. Furthermore, $w_2 = v_1w_1v_2$, $v_1v_2 \neq e$, and $x = x_1w_2{}^kx_2$ where $\langle v\rangle \Rightarrow^* x_1v^kx_2$ for any v in L_A. If w_1 contains two or more letters, no duplications occur in the derivations from $\langle w_1\rangle$ and $\langle w_2\rangle$ and $k = 1$. Then $\{uu_1u_2{}^nw_1v_2{}^nx_2z \mid n \geqslant 1\}$ is a subset of L for $v_1v_2 \neq e$, which is impossible.

If $w_2 = a_i{}^s$, then $w_1 = a_i{}^r$ and $1 \leqslant r \nleqslant s$. In this case L contains both $uu_1x_1a_i^{rk}x_2z$ and $uu_1u_2x_1a_i^{sk}x_2z$. If uu_1 or a_ix_2 contain two or more letters, the middle of the string is determined, so $u_2 x_1 a_i^{sk} = x_1 a_i^{rk}$, which is impossible since the words have different lengths; otherwise, $x_1 a_i^{rk}$ contains three or more symbols, so $uu_1 = uu_1u_2$ and $x_2 z = a_i^{sk-rk} x_2 z$, which is also impossible.

Thus if w_1 contains two or more letters or w_2 contains only one letter, we obtain a contradiction. Notice that once the bracketed portion of a derivation has two or more letters, this condition persists. Hence if there is a derivation from $\langle w \rangle$ of length $2 \mid V_M \mid + 3$ or greater, then (1) must occur with w_1 containing one letter. Thus derivations from $\langle w \rangle$ cannot be longer than $2 \mid V_M \mid + 2$. This contradicts the fact that $L\langle w \rangle$ is infinite. $\square$

Corollary 1. $W_{RB} - W_R \neq \varnothing$.

Proof. Example 4 shows that L is in W_{RB}.

Remark. The argument above shows that if f_1, f_2, f_3, and f_4 are strictly increasing functions from the positive integers into the positive integers, then W_R cannot contain any infinite subset of $\{a_1^{f_1(n)} a_2^{f_2(n)} a_3^{f_3(n)} a_4^{f_4(n)} \mid n \geqslant 1\}$.

If we add one more coordinate, a_5, we can use the arguments of Theorem 5.1 to show that the resulting language is not in W_{RB}.

Theorem 5.2. The language $\hat{L} = \{a_1{}^n a_2{}^n a_3{}^n a_4{}^n a_5{}^n \mid n \geqslant 1\}$ is not in W_{RB}.

Proof. We adopt the argument in Theorem 5.1. Note that our remarks on the structure of L also apply to $\hat{L}$ and in addition if $\{u_1 v_1{}^n u_2 v_2{}^n u_3 v_3{}^n u_4 v_4{}^n u_5 \mid n \geqslant 1\} \subseteq \hat{L}$, then $v_1 = v_2 = v_3 = v_4 = e$.

If $\hat{L}$ is in W_{RB}, we can assume $\hat{L} = L(G)$ for a reduced, factored, lossless, normal, regular-based W-grammar $G = (V_M, V_P, \Sigma, P_M, P_h, \sigma)$ obtained using the constructions of Theorems 3.1 and 3.2.

First we notice that at most one variable per hyperrule can be "usable." Suppose $\sigma \Rightarrow^* u_1\langle w_1 \rangle\, u_2 \langle w_2 \rangle u_3$ for $u_1, u_2, u_3 \in \Sigma^*$, $w_1, w_2 \in V_P{}^+$. Then $u_1 L\langle w_1 \rangle\, u_2 L\langle w_2 \rangle u_3 \subseteq \hat{L}$ so either $\mid L\langle w_1 \rangle \mid = 1$ or $\mid L\langle w_2 \rangle \mid = 1$. If $x \in L\langle w \rangle$ and $w \in L_\beta$, there is a factorization $x = x_1 w \cdots x_k w x_{k+1}$ such that $x_1 w' \cdots x_k w' x_{k+1}$ is in $L\langle w' \rangle$ for all w' in L_β. Hence if $L\langle w \rangle = \{x\}$, we could replace $\langle \beta \rangle$ in a hyperrule by $x_1 \beta \cdots x_k \beta x_{k+1}$. So if G has a hyperrule $\gamma \to y_1 \langle \beta_1 \rangle\, y_2 \langle \beta_2 \rangle\, y_3$, there is either a string $\alpha \in (V_M \cup V_P)^+$ such that we can replace the hyperrule either by $\gamma \to y_1 \langle \beta_1 \rangle\, y_2 \alpha y_3$ or by $\gamma \to y_1 \alpha y_2 \langle \beta_2 \rangle\, y_3$. Similarly, if $\sigma \Rightarrow^* u_1 Z u_2 Y u_3$ for $Z, Y \in V_P - \Sigma$, either $\mid L(Z) \mid = 1$ or $\mid L(Y) \mid = 1$ and analogously for $\sigma \Rightarrow^* u_1 Z u_2 \langle w \rangle u_3$ or $\sigma \Rightarrow^* u_1 \langle w \rangle\, u_2 Z u_3$.

We can eliminate protovariables and independent metavariables from hyperrules, using arguments analogous to those in the proof of Theorem 5.1. We can finally conclude that the hyperrules of G are in the forms

$$\sigma \to w, \qquad \sigma \to u\langle w \rangle v, \qquad \langle A \rangle \to u_1 \langle v_1 A v_2 \rangle u_2, \qquad \langle A \rangle \to u_1 A u_2$$

for $w \in \Sigma^+$; $u, v \in \Sigma^*$; $u_1, u_2 \in (\Sigma \cup \{A\})^*$; $v_1 v_2 \in (\Sigma \cup \{A\})^+$.

Again we observe that we must have $L\langle w\rangle$ infinite for some $w \in \Sigma^+$ with $\sigma \to u\langle w\rangle v$ in P_h. Hence there must be arbitrarily long derivations from $\langle w\rangle$ and so derivations of at least $2 \mid V_M \mid + 3$ steps. But such a derivation can be divided:

$$\langle w\rangle \stackrel{*}{\Rightarrow} u_1\langle w_1\rangle v_1 \stackrel{*}{\Rightarrow} u_1u_2\langle w_2\rangle v_2v_1 \stackrel{*}{\Rightarrow} u_1u_2xv_2v_1 \tag{2}$$

where u_1, u_2, w_1, w_2, z_1, z_2, $x \in \Sigma^*$, w_1, $w_2 \in L_A$ for some A, and either $w_2 = a_i{}^s$ or w_1 contains two or more letters. We can factor $x = x_1w_2{}^kx_2$ so that $\langle v\rangle \Rightarrow^* x_1z^kx_2$ for each $z \in L_A$.

If $w_2 = a_i{}^s$, then $w_1 = a_i{}^r$ for $1 \leqslant r \nleqslant s$ and both $uu_1x_1a_i^{kr}x_2v_1v$ and $uu_1u_2x_1a_i^{ks}x_2v_1v$ are in $\hat{L}$. To avoid the sort of contradiction we obtained before, we must have $uu_1 \in a_1{}^*$, $v_1v \in a_5{}^*$, $i = 3$, $x_1 \in a_2{}^+a_3{}^*$, and $x_2 \in a_3{}^*a_4{}^+$. We must have $u_2 \in a_1{}^+a_2{}^+$ and $v_2 \in a_4{}^+a_5{}^+$ to balance the increase in the a_3's. But since $w_2 = a_3{}^s \in L_A$ and any duplications in $\langle a_3{}^r\rangle \Rightarrow^* u_2\langle a_3{}^s\rangle v_2$ must take place wholly *within* brackets, we have $\langle a_3{}^s\rangle \Rightarrow^* u_2\langle a_3{}^t\rangle v_2$ for some $a_3{}^t \in L_A$. But then $uu_1u_2u_2x_3a_1^{kt}x_2v_2v_2v_1v$ is in $\hat{L}$, an obvious contradiction.

On the other hand, if w_1 contains two or more letters, no duplications occurred in $\langle w_1\rangle \Rightarrow^* u_2\langle w_2\rangle v_2$ or $\langle w_2\rangle \Rightarrow^* x_1w_2x_2$, $k = 1$, and $w_2 = z_1w_1z_2$, $v_1v_2 \neq e$. Then we have

$$\{uu_1u_2{}^nx_1z_1{}^nw_1z_2{}^nx_2v_2{}^nv_1v \mid n \geqslant 1\} \subseteq \hat{L}$$

a contradiction.

Corollary 1. $W_{RB} \subsetneq \mathscr{Q}$.

Corollary 2. W_{RB} is properly contained in the family of index languages.

Proof. The language L is clearly generated by the macroproductions $S \to F(a_1, a_2, a_3, a_4, a_5)$,

$$F(x_1, x_2, x_3, x_4, x_5) \to F(a_1x_1a_2x_2a_3x_3a_4x_4a_5x_5)$$

and

$$F(x_1x_2x_3x_4x_5) \to x_1x_2x_3x_4x_5 \quad \square$$

We conclude our discussion of normal regular-based W-grammars with some brief comments on their closure properties. Roughly speaking, W_{RB} is closed under operations involving symbol replacement but not under those depending on the order of symbols. Thus W_{RB} is closed under nonerasing homomorphism, substitution, and nested iterated substitution but not under intersection with regular sets.

Definition 5.1. A homomorphism h is *nonerasing* if $h(w) \neq e$ for $w \neq e$.

Definition 5.2. Let Σ be a finite vocabulary. A *substitution* τ on Σ^* is defined by associating a language $\tau(a) = L_a$ to each $a \in \Sigma$, extending τ to Σ^* by $\tau(e) = \{e\}$ and $\tau(xy) = \tau(x)\,\tau(y)$ for $x, y \in \Sigma^*$, and finally for $L \subseteq \Sigma^*$, letting $\tau(L) = \bigcup_{w \in L} \tau(w)$. The substitution τ is *nonerasing* if $e \notin \tau(a)$ for each $a \in \Sigma$; it is *nested* if $a \in \tau(a)$ for each $a \in \Sigma$. A family of languages $\mathscr{L}$ is *closed under nonerasing substitution* if whenever $L \in \mathscr{L}$, $L \subseteq \Sigma^*$, and τ is a nonerasing substitution with $\tau(a) \in \mathscr{L}$ for each $a \in \Sigma$, then $\tau(L) \in \mathscr{L}$.

Definition 5.3. Let Σ be a finite vocabulary and τ a substitution on Σ^*. If $\tau(a) \subseteq \Sigma^*$ for each $a \in \Sigma$, and $L \subseteq \Sigma^*$, let $\tau^0(L) = L$, and $\tau^{n+1}(L) = \tau(\tau^n(L))$ for $n \geqslant 0$. We call $\tau^\infty(L) = \bigcup_{n \geqslant 1} \tau^n(L)$ an *iterated substitution* and if τ is (nonerasing) nested, then τ^∞ is a (nonerasing) *nested iterated substitution*. A family of languages $\mathscr{L}$ is *closed under* (*nonerasing*) *nested iterated substitution* if whenever $L \in \mathscr{L}$, $L \subseteq \Sigma^*$, and τ is a (nonerasing) nested substitution with $\tau(a) \subseteq \Sigma^*$ and $\tau(a) \in \mathscr{L}$ for each $a \in \Sigma$, then $\tau^\infty(L)$ is in $\mathscr{L}$.

Theorem 5.3. The family W_{RB} is closed under nonerasing substitution and nonerasing, nested, iterated substitution but not under intersection with regular sets.

Proof. Let $L \in W_{RB}$, $L \subseteq \Sigma^+$, and let τ be a nonerasing substitution such that $\tau(a) \in W_{RB}$ for each $a \in \Sigma$. We wish to use the same sort of construction as for context-free grammars. However, we must use distinct terminals as well as distinct variables in order to prevent using a production available for $\tau(a)$ while generating a word in $\tau(b)$ for $a \neq b$. So we "paint" $\tau(a)$ and $\tau(b)$ different "colors."

For $a, b \in \Sigma$ (a, b may be equal or unequal) let $(a, 1)$ and (a, b) be new symbols. Let h and h_b be homomorphisms given by $h(a) = (a, 1)$ and $h_b(a) = (a, b)$ for $a \in \Sigma$. Let $\Sigma_b = \{(a, b) \mid a \in \Sigma\}$ and $\Sigma_1 = \{(a, 1) \mid a \in \Sigma\}$. Now renaming all symbols with new and distinct names obviously does not affect membership in W_{RB}. Hence the language $L_1 = h(L)$ is in W_{RB}, as are the languages $L_a = h_a(\tau(a))$ for each $a \in \Sigma$.

So we may assume we have normal regular-based W-grammars

$$\begin{aligned} G_1 &= (V_{M,1}, V_{P,1}, \Sigma_1, P_{M,1}, P_{h,1}, \sigma_1), \\ G_a &= (V_{M,a}, V_{P,a}, \Sigma_a, P_{M,a}, P_{h,a}, \sigma_a) \end{aligned}$$

such that $L_1 = L(G_1)$, $L_a = L(G_a)$ for $a \in \Sigma$ and

$$(V_{M,1} \cup V_{P,1}) \cap (V_{M,a} \cup V_{P,a}) = (V_{M,a} \cup V_{P,a}) \cap (V_{M,b} \cup V_{P,b}) = \varnothing$$

for $a, b \in \Sigma$, $a \neq b$.

To obtain $\bar{G}$ such that $L(\bar{G}) = \tau(L)$, we let $\bar{G}$ have as metaproductions

$$\bar{P}_M = P_{M,1} \bigcup_{a \in \Sigma} P_{M,a}$$

and as hyperrules

$$\bar{P}_h = P_{h,1} \bigcup_{a \in \Sigma} P_{h,a} \cup \{(a, i) \to \sigma_a \mid a \in \Sigma\} \cup \{(a, b) \to a \mid a, b \in \Sigma\}$$

Then

$$\bar{V}_M = V_{M,1} \bigcup_{a \in \Sigma} V_{M,a}, \qquad \bar{V}_P = V_{P,1} \bigcup_{a \in \Sigma} V_{P,a} \cup \Sigma$$

and

$$\bar{G} = (\bar{V}_M, \bar{V}_P, \Sigma, \bar{P}_M, \bar{P}_h, \sigma_1)$$

If τ is nested, then we obtain $\hat{G}$ with $L(\hat{G}) = \tau^{\infty}(L)$ by selecting as hyperrules

$$\hat{P}_h = \bar{P}_h \cup \{(a, 1) \to a \mid a \in \Sigma\} \cup \{(a, b) \to \sigma_a \mid a, b \in \Sigma\}$$

we let $\hat{G} = (\bar{V}_M, V_P, \Sigma, \bar{P}_M, \hat{P}_h, \sigma_1)$.

We show that W_{RB} is not closed under intersection with regular sets by recalling that $L = \{a_1{}^n a_2{}^n a_3{}^n a_4{}^n a_5{}^n \mid n \geqslant 1\}$ is not in W_{RB}. On the other hand, let $\Sigma = \{a_1, a_2, a_3, a_4, a_5\}$ and

$$L = \{x_1 a_1 x_2 a_2 x_3 a_3 x_4 a_4 x_5 a_5 \mid \exists n \geqslant 0, \quad \forall i, \quad |x_i| = n, \quad x_i \in \Sigma^*\}$$

and $R = a_1{}^+ a_2{}^+ a_3{}^+ a_4{}^+ a_5{}^+$. Then $L = L_1 \cap R$ and R is regular. But L_1 can be generated by a normal regular-based W-grammar with metavariable N, protovariables γ and Z, metaproductions $N \to Z$, $N \to ZN$, and hyperrules

$$\sigma \to \langle N \rangle, \qquad \langle N \rangle \to N a_1 N a_2 N a_3 N a_4 N a_5, \qquad \sigma \to a_1 a_2 a_3 a_4 a_5$$

and

$$Z \to a_i \quad \text{for} \quad 1 \leqslant i \leqslant 5 \quad \square$$

6. SIMPLE W-GRAMMARS

A W-grammar is *simple* if it is *strict*—each $L_A \subseteq \Sigma^+$—and it is *unary*—whenever $\langle \beta \rangle$ appears anywhere in a hyperrule then $\beta = A$ for a metavariable A. Let W_S be the set of all languages generated by simple W-grammars. It is evident that W_S -like W_{RB}—properly contains the family of context-free languages; we shall show that W_{RB} is properly contained in $\mathcal{Q}$. We shall also see that W_{RB} and W_S are incomparable—$W_{RB} - W_S \neq \varnothing$ and $W_S - W_{RB} \neq \varnothing$. Finally, we notice that W_S has no interesting closure properties.

First we observe that W_S contains every language expressible as the intersection of two context-free languages. This easy result contrasts with the surprising fact that W_S does *not* contain certain languages expressible as the intersection of *three* context-free languages!

Theorem 6.1. If L_1 and L_2 are context-free, then $L_1 \cap L_2 \in W_S$.

Proof. Let $L_1 = L(G_1)$ and $L_2 = L(G_2)$ for context-free grammars $G_1 = (V_1, \Sigma, P_1, S_1)$ and $G_2 = (V_2, \Sigma, P_2, S_2)$ with $V_1 \cap V_2 = \Sigma$. Let σ be a new symbol and $P_h = \{\sigma \to \langle S_1 \rangle, \langle S_2 \rangle \to S_2\}$. Then $L = L(G)$ for the simple W-grammar

$$G = ((V_1 - \Sigma) \cup (V_2 - \Sigma), \{\sigma\} \cup \Sigma, \Sigma, P_1 \cup P_2, P_h, \sigma) \quad \square$$

Corollary 1. The family of context-free languages is properly contained in W_S.

Corollary 2. W_S contains languages that are not indexed languages.

Proof. The family of indexed languages does not contain every language of the form $L_1 \cap L_2$ for L_1 and L_2 context-free.[1] $\square$

Corollary 3. The language $L = \{a_1{}^n a_2{}^n a_3{}^n a_{4n} a_5{}^n \mid n \geqslant 1\}$ is in $W_{RB} - W_S$.

Proof. The language L is expressible as the intersection of two context-free languages. $\square$

We shall show that $W_S \subsetneq \mathscr{Q}$ but the containment is *not* effective. For each simple W-grammar G there is a nondeterministic multitape Turing machine M accepting $L(G)$ in realtime, but there is *no* algorithm to construct M from G. Indeed, although each language in W_S is recursive, each simple W-grammar is not recursive in the following sense.

Theorem 6.2. The question "is w in $L(G)$" is undecidable for simple W-grammars G.

Proof. For context-free grammars G_1 and G_2 the question "$L(G_1) \cap L(G_2) = \varnothing$" is undecidable.[5] We can assume that $G_1 = (V_1, \Sigma, P_1, S_1)$ and $G_2 = (V_2, \Sigma, P_2, S_2)$ where $V_1 \cap V_2 = \Sigma$. Let σ, a be new symbols. We can construct from G_1 and G_2 a set of hyperrules $P_h = \{\sigma \to \langle S_1 \rangle, \langle S_2 \rangle \to a\}$ and a simple W-grammar

$$G = ((V_1 - \Sigma) \cup (V_2 - \Sigma), \{\sigma\} \cup \Sigma, \Sigma, P_1 \cup P_2, P_h, \sigma)$$

Then $a \in L(G)$ if and only if $L(G_1) \cap L(G_2) \neq \varnothing$. Thus if membership were decidable for simple W-grammars, emptiness would be decidable for $L(G_1) \cap L(G_2)$. $\square$

Despite Theorem 6.2, it is true that $W_S \subseteq \mathscr{Q}$. If we have a simple W-grammar $G = (V_M, V_P, \Sigma, P_M, P_h, \sigma)$ and let $\pi(S) = \bigcap_{A \in S} L_A$ for

$S \subseteq V_M$, then if we were given the set $\mathscr{S}_G = \{S \subseteq V_M \mid \pi(S) \neq \varnothing\}$, we could construct a nondeterministic multitape Turing machine to accept $L(G)$ in realtime. Unfortunately, there is no algorithm to locate $\mathscr{S}_G$ given G.

Let us see what could be done if a "birdie" told us the membership of $\mathscr{S}_G$.

Theorem 6.3. $W_S \subseteq \mathscr{Q}$. Given a simple W-grammar $G = (V_M, V_P, \Sigma, P_M, P_h, \sigma)$ and the set $\mathscr{S}_G = \{S \subseteq V_M \mid \bigcap_{A \in S} L_A \neq \varnothing\}$, one can construct a nondeterministic multitape Turing machine to accept $L(G)$ in realtime. If G is lossless, it is not necessary to know $\mathscr{S}_G$.

Proof. We only outline the necessary steps. As in the proof that $W_{RB} \subseteq \mathscr{Q}$, it suffices to construct a nondeterministic multitape Turing machine M to generate $L(G)$ in linear time—i.e., there is an integer k such that if $w \in L(G)$, M generates W in at most $k \mid w \mid$ steps.

The machine M has as before a tape TL for the left part of the derived string, a tape TR for the right part, a tape TC for the current protovariable or hypernotion being expanded, and up to $\mid V_M \mid$ auxiliary two-headed tapes for duplicating, and up to $\mid V_M \mid$ extra pushdown store tapes to check membership in each $\pi(S)$. On TC, M will have either Z for a protovariable Z, or $\langle S, w\rangle$ for $S \in \mathscr{S}_G$ and $w \in \pi(S)$ or $\bar{S}$ for $S \in \mathscr{S}_G$.

Whenever M wants to apply a production $\gamma \rightarrow y$ with $y \in \Sigma^+$ it again guesses whether TL has any substrings to be expanded (protovariables or hypernotions). If it has none, M puts y on the right of TL and transfers terminals from the left of TR to the right of TL until it hits a protovariable or hypernotion η, whereupon it places a barrier $\square$ on TL and puts η on TC. If M guesses that TL has a substring η to expand, it transfers terminals from the right of TL to the left of TR until either it finds η and puts η on TC or it hits a barrier or empties TL, in which cases M blocks.

Suppose M has a protovariable Z on TC, and selects a hyperrule

$$Z \rightarrow u_0\langle A_1\rangle \cdots u_{m-1}\langle A_m\rangle u_m$$

If $m = 0$ and $u_0 \in (\Sigma \cup V_M)^+$, M selects a corresponding production $Z \rightarrow u_0'$ for $u_0' \in \Sigma^+$ and behaves as above. If $m = 0$ but u_0 contains a protovariable Y and the corresponding production is $Z \rightarrow u_0' Y u_0''$, M writes u_0' on TL, u_0'' on TR, and replaces Z by Y on TC. If $m \geqslant 1$, M selects a metaassignment h, puts $h(u_0)$ on TL, γ_1 on TC, and $h(u_1)\gamma_2 \cdots h(u_{m-1})\, \gamma_m u_m$ on TL where for each i either $\gamma_i = \bar{S}_i$, $A_i \in S_i$, $h(A_i) \in \pi(S_i)$, and $S_i \in \mathscr{S}_G$, or else $\gamma_i = \langle S_i, h(A_i)\rangle$, $h(A_i) \in \pi(S_i)$, $A_i \in S_i$, and $S_i \in \mathscr{S}_G$; if $A_i = A_j$, then $S_i = S_j$. Notice that if A_i does not appear in any u_j and whenever $A_j = A_i$ then $\gamma_i = \bar{S}_i$, there is no need to compute $h(A_i)$; it suffices to select any $S_i \in \mathscr{S}_G$ with $A_i \in S_i$. Also, each $\pi(S)$ is quasirealtime[8] so $\langle S_i, h(A_i)\rangle$ can be written, suitably encoded, in time $\mid h(A_i)\mid$.

If TC contains $\langle S, w\rangle$, M selects a lossless hyperrule

$$\langle A\rangle \rightarrow u_0\langle A_1\rangle \cdots u_{m-1}\langle A_m\rangle u_m$$

(A appears on the right) with $A \in S$ and a metaassignement h such that $h(A) = w$. If $m = 0$, M behaves as above. If $m = 1$, there are three cases. Suppose A is one of the A_i ; M selects any r with $A_r = A$. If A does not appear in any u_j, M places $h(u_0)\, \gamma_1 \cdots h(u_{r-1})$ on the left of TL, leaves $\gamma_r = \langle S, w\rangle$ on TC, and puts $h(u_r)\, \gamma_{r+1} \cdots h(u_{m-1})\, \gamma_m h(u_m)$ on TR, where the γ_i are selected as before; on the other hand, if $A = A_r$ and A appears in some u_j, M either behaves as in the previous case, or puts $h(u_0)$ on TL, γ_1 on TC, and $h(u_1) \cdots h(u_m)$ on TR where in this case $\gamma_j = \bar{S}_r$ whenever $A_j = A = A_r$. If $A_i \neq A$ for all A but A appears in some u_j, M puts $h(u_0)$ on TL, γ_1 on TC, and $h(u_1) \cdots h(u_m)$ on TR where the γ_i are selected as before.

If $\bar{S}$ appears on TC for $S \in \mathscr{S}_G$, M selects a hyperrule

$$\langle A\rangle \rightarrow u_0\langle A_1\rangle \cdots u_{m-1}\langle A_m\rangle u_m$$

with $A \in S$ such that A does not appear in any u_j. If $m = 0$, M behaves as before. If $m \geqslant 1$, M selects a metaassignment h. If $A_r = A$, M leaves $\gamma_r = \bar{S}$ on TC, puts $h(u_0)\, \gamma_1 \cdots h(u_{r-1})$ on TL and $h(u_r)\, \gamma_{r+1} \cdots h(u_m)$ on TR, where now $\gamma_j = \bar{S}$ whenever $A_j = A = A_r$. If $A_i \neq A$ for all i, M puts $h(u_0)$ on TL, γ_1 on TC, and $h(u_1) \cdots h(u_m)$ on TR.

The important point here is that we use $\bar{S}$ as a new protovariable substituting for $\langle A\rangle$ when $A \in S$ and we guess that no production subsequently applied to $\langle h(A)\rangle$ will "deposit" $h(A)$ in unbracketed form. In such a situation we do not need to know $h(A)$—just the hyperrules applicable to $\langle h(A)\rangle$ and this information is given to us by $\bar{S}$. When we use $\bar{S}$ instead of $\langle h(A)\rangle$ we say that $h(A) \in L_B$ for all $B \in S$ and guess that we will never use a hyperrule starting with $\langle B\rangle$ for $B \notin S$. Thus in effect we treat G as if it were lossless. Since we can verify a guess that $w \in \pi(S)$ in the time it takes to write $\langle S, w\rangle$, the same argument as before (Theorem 4.2) shows that M operates in linear time. $\square$

Now we examine some of the limits on simple W-grammars. On the one hand, simple W-grammars can generate the intersection of two context-free languages and so can generate languages which are not indexed languages.(1) On the other hand, we shall see next that certain subsets of a^*—such as $\{a^{n^2} \mid n \geqslant 1\}$ and $\{a^{2^{2^n}} \mid n \geqslant 1\}$—are in W_{RB} and even W_R but not W_S.

Theorem 6.4. Let $L \subseteq a^*$ be in W_S. If L is infinite, it contains an infinite regular set.

Proof. Let $L = L(G)$ for the simple W-grammar $G = (V_M, V_P, \Sigma, P_M, P_h, \sigma)$ and assume $L \subseteq a^*$. First, since the context-free languages are closed under intersection with regular sets, we can assume that for each $A \in V_M$ either $L_A \subseteq a^+$ or else $L_A \subseteq \Sigma^*(\Sigma - \{a\})\,\Sigma^*$. Second, if $L_A \subseteq \Sigma^*(\Sigma - \{a\})\,\Sigma^*$, then for any hyperrule $\gamma \to u_0\langle A_1\rangle \cdots u_{m-1}\langle A_m\rangle u_m$ either the rule is never used or A cannot appear in any u_j. Thus $\langle A\rangle$ could be replaced by protovariables $\bar{S}_A$ for $A \in S_A$, $S_A \subseteq V_M$, and $S_A \in \mathscr{S}_G$ where $\mathscr{S}_G$ is as before. Hence there is (but we may not be able to find it) an equivalent W-grammar with each $L_A \subseteq a^+$. But a context-free subset of a^* is regular.[14]

Since we are only interested in an existence proof, we may as well assume that G is regular-based as well as strict and unary and so, by Theorems 3.1 and 3.2, also factored, lossless, and reduced.

Suppose that L does not contain an infinite regular set. If A is a metavariable appearing independently in a hyperrule, the arguments of Theorems 5.1 and 5.2 show that $L = L(G)$ contains a subset of the form

$$w_1 \operatorname{Dup}(L_A, k) w_2$$

for $w_1, w_2 \in \Sigma^*$, $k \geqslant 1$. Since $L_A \subseteq a^+$ and L_A is infinite, $w_1 \operatorname{Dup}(L_A, k) w_2$ is an infinite regular subset of L. Hence, we can assume that there are no creative hyperrules.

So for any n, G contains only finitely many derivations of length n or less. If $\sigma \Rightarrow^* x\langle w\rangle y$, $\langle w\rangle \Rightarrow^* u\langle w\rangle v$, and $\langle w\rangle \Rightarrow^* z$ for $xy \in a^*$; $w, uv, z \in a^+$; then L contains the infinite regular subset $xy(uv)^* z$. If $\sigma \Rightarrow^* xZy$, $Z \Rightarrow^* uZv$, and $Z \Rightarrow^* w$ for $Z \in V_P - \Sigma$; $x, y \in a^*$; $w, uv \in a^+$; then L contains the infinite regular subset $xy(uv)^* w$. Finally, if $\sigma \Rightarrow^* x\langle w_1\rangle y$, $\langle w_1\rangle \Rightarrow^* u\langle w_2\rangle v$, and $\langle w_2\rangle \Rightarrow^* w$ with $w_1 \neq w_2$; $x, y \in a^*$; $uv, w_1, w_2, w \in a^+$; and $w_1, w_2 \in L_A$; then there is a factorization $u = u_1 w_1{}^k u_2$, $v = v_1 w_1{}^l v_2$ such that $\langle z\rangle \Rightarrow^* u_1 z^k u_2\langle w_2\rangle\, v_1 z^l v_2$ for all $z \in L_A$. Hence $\langle w_2\rangle \Rightarrow^* u_1 w_2{}^k u_2\langle w_2\rangle\, w_1 w_2{}^l v_2$ and so L contains the infinite regular set $xuvw(u_1 w_2{}^k u_2 v_1 w_2{}^l v_2)^*$. Thus no complete derivation from σ can contain a path of length greater than $|\, V_M \cup V_P\,| + 1$, so there are only finitely many complete derivations. Hence $L = L(G)$ is finite. $\square$

Corollary 1. W_{RB} and W_S are incomparable.

Proof. By Corollary 3 to Theorem 6.1, $W_S - W_{RB} \neq \varnothing$. On the other hand, the language $L = \{a^{n^2} \mid n \geqslant 1\}$ is in W_{RB} (and even W_R) by Example 2 but is not in W_S. $\square$

Corollary 2. W_S is not closed under nonerasing homomorphism.

Proof. Let

$$L_1 = c\{a^n c a^{n+2} c \mid n \geqslant 1\}^* \cup c\{a^n c a^{n+2} c\}^*\, a^+ c$$

and

$$L_2 = \{c\} \cup caac\{a^n c a^{n+2} c \mid n \geqslant 1\}^* \cup caac\{a^n c a^{n+2} c \mid n \geqslant 1\}^*\, a^+ c$$

and $h(a) = h(c) = a$. Then L_1 and L_2 are context-free, so $L_1 \cap L_2$ is in W_S but $h(L_1 \cap L_2) = \{a^{n^2} \mid n \geqslant 1\}$ is not in W_S. $\square$

We now show that W_S does not contain all intersections of context-free languages.

Theorem 6.5. The language

$$L = \{a_1{}^n a_2{}^m a_3{}^k b_1{}^n b_2{}^m b_3{}^k c_1{}^n c_2{}^m c_3{}^k \mid n, m, k \geqslant 1\}$$

is not in W_S.

Proof. Suppose $L = L(G)$ for a simple W-grammar $G = (V_M, V_P, \Sigma, P_M, P_h, \sigma)$. The first part of our proof hinges on structural facts about L similar to those employed in Theorems 5.1 and 5.2. First, L does not contain any infinite context-free language. Second, if any three of $u_1w_1v_1$, $u_1w_2v_1$, $u_2w_1v_2$, and $u_2w_2v_2$ are in L, either $w_1 = w_2$ or $u_1 = u_2$ and $v_1 = v_2$. We shall use these facts to narrow down the possibilities until we get L expressed as the union of intersections of two context-free languages, and then use a result of Liu and Weiner.[16]

The arguments in the proofs of Theorems 5.1 and 5.2 show that we can assume that G contains no protovariables except σ and σ never appears on the right-hand side of a hyperrule. If a complete derivation has a subpart $\langle w\rangle \Rightarrow^* u\langle w\rangle v$ for $uv \in \Sigma^+$, then $L(G)$ contains an infinite context-free language, a contradiction. So we can assume that G has no hyperrule of the form $\langle A\rangle \to u\langle A\rangle v$. This is the key observation.

The role of a hypernotion $\langle w\rangle$ in a derivation depends on only two things. It must arise from a hyperrule $\langle C\rangle \to u\langle A\rangle v$ with $A \neq C$, i.e., A independent and w in L_A. Then the rule applied to $\langle w\rangle$ comes from a hyperrule $\langle B\rangle \to u_0\langle A_1\rangle \cdots u_{m-1}\langle A_m\rangle u_m$ where $A_i \neq B$ for all i and $w \in L_B$. Then $\langle w\rangle$ disappears. We call A the first metavariable of $\langle w\rangle$ and B the second. For any other w' in $L_A \cap L_B$, $\langle w'\rangle$ could play the role of $\langle w\rangle$ in that derivation.

Suppose $\sigma \Rightarrow^* x\langle w_1\rangle y$; $\langle w_1\rangle \Rightarrow^* u\langle w_2\rangle v$; $\langle w_2\rangle \Rightarrow^* w$; $x, y \in \Sigma^*$; uv, $w \in \Sigma^+$; and both $\langle w_1\rangle$ and $\langle w_2\rangle$ have first metavariable A and second metavariable B. Then $\langle w_1\rangle$ arose from a hyperrule $\langle C\rangle \to \alpha\langle A\rangle\beta$, $A \neq C$, and a metaassignment h with $h(A) = w_1$, yielding a production, $\langle h(C)\rangle \to h(\alpha)\langle w_1\rangle\, h(\beta)$. If we let $h_1(A) = w_2$ and $h_1(D) = h(D)$ for $A \neq D$, then $\langle h(C)\rangle \to h_1(\alpha)\langle w_2\rangle\, h_1(\beta)$ is also a production. If $\langle w_2\rangle$ appears in $h_1(\alpha)$ or $h_1(\beta)$, we can derive w from it and otherwise proceed as in the original derivation. Hence, there are $x', y' \in \Sigma^*$ with $\sigma \Rightarrow^* x'\langle w_2\rangle y'$. Similarly there are u', v' with $u'v' \in \Sigma^*$ such that $\langle w_2\rangle \Rightarrow^* u'\langle w_2\rangle v'$. Thus $L(G)$ would contain an infinite context-free language.

We can likewise argue that if $\langle w_1\rangle \Rightarrow \langle w_2\rangle \Rightarrow \cdots \Rightarrow \langle w_n\rangle$, we can assume that $n \leqslant |V_M|^2 + 2$, for otherwise there is a shorter derivation from $\langle w_1 \rangle$ to $\langle w_n \rangle$. Hence we can assume that no derivation tree has a path longer than $p^2 + p + 1$, where $p = |V_M'|^2 + 2$.

We can obtain from a complete derivation in G a skeleton derivation tree T as follows. The root of T is labeled σ. Instead of a node labeled $\langle w \rangle$ we have $\langle A, B \rangle$ where A and B are the first and second metavariables of w, respectively. If the production applied to $\langle w \rangle$ came from a hyperrule $\langle B \rangle \to u_0 \langle A_1 \rangle \cdots u_{m-1} \langle A_m \rangle u_m$ and $u_i = Z_{i1} \cdots Z_{ir_i}$, each $Z_i \in V_M \cup \Sigma$, then $\langle A, B \rangle$ has $m + \Sigma_{i=0}^m r_i$ sons labeled $Z_{01}, \ldots, Z_{0r_0}, \langle A_1, B_1 \rangle, Z_{11}, \ldots$, etc., for appropriate B_i. The nodes labeled Z_{ij} are, of course, leaves.

If a leaf labeled B has a father labeled $\langle A, B \rangle$, then no brother can be labeled $\langle B, A_1 \rangle$. In this case, to form a member of $L(G)$, B can be replaced by all and only members of $L_A \cap L_B$ provided all brothers labeled B and uncles labeled A are replaced by the same member of $L_A \cap L_B$. Similarly, if a leaf labeled B has a brother labeled $\langle B, A \rangle$, it can be replaced by all and only members of $L_A \cap L_B$ provided the same replacement is made to any brother labeled B and to any son of $\langle B, A \rangle$ labeled A. Now in all these cases if members of $L_A \cap L_B$ can be duplicated, then $L_A \cap L_B \subseteq a^+$ for some letter $a \in \Sigma$ and the same is true for anything lying in between these duplications. Thus we can rearrange leaves so that a leaf labeled B is adjacent to all its brothers labeled B; we can also move the duplicated "uncles" or "nephews" next to B and replace the whole mess by one leaf labeled $\mathrm{Dup}(L_A \cap L_B, k)$ for an appropriate k.

If a leaf labeled B has no father labeled $\langle A, B \rangle$ or brother labeled $\langle B, A \rangle$, it and its brothers labeled B can be replaced by any member of $L_B = L_B \cap L_B$. Again they can be collected and replaced by $\mathrm{Dup}(L_B \cap L_B, k)$.

If we read off the labels on the leaves of T, we now get a language $L_T \subseteq L(G)$ of the form

$$L_T = w_0 \,\mathrm{Dup}(L_1 \cap L_1', k_1) \cdots w_{m-1} \,\mathrm{Dup}(L_m \cap L_m', k_m) w_m \subseteq L(G)$$

where $w_i \in \Sigma^*$, $k_i \geqslant 1$, and each L_i, L_i' is context-free. First, notice that if $k_i \geqslant 2$, then $L_i \cap L_i' \subseteq a^*$ for some $a \in \Sigma$, and since $L_i \cap a^+$ is regular, $L_i \cap L_i'$ and so $\mathrm{Dup}(L_i \cap L_i', k_i)$ is regular and hence context-free. Thus we can in effect assume L_T is expressed as

$$L_T = w_0(L_1 \cap L_1') \cdots w_{m-1}(L_m \cap L_m')w_m$$

for L_i, L_i' context-free.

Next, notice that $w_i(L_{i+1} \cap L_{i+1}') = (w_i L_{i+1} \cap w_i L_{i+1}')$ and each $w_i L_{i+1}$, $w_i L_{i+1}'$ is context-free. Thus we can assume L_T is expressed as

$$L_T = (L_1 \cap L_1') \cdots (L_m \cap L_m')$$

each L_i, L_i' is context-free. Finally, notice that we must have $|L_i \cap L_i'| = 1$ for all but one i. So we can assume that L_T is the intersection of two context-free languages.

Now we saw that we can assume that no tree has a path of length more than $p^2 + p + 1$. Hence there are finitely many trees $T_1, \ldots, T_q$ such that

$$L(G) = \bigcup_{i=1}^{q} L_{T_i}, \qquad L_{T_i} = L_{i1} \cap L_{i2}$$

for L_{i1}, L_{i2} context-free, $1 \leqslant i \leqslant q$.

Now Liu and Weiner[16] have shown that

$$\{a_1^{n_1} a_2^{n_2} a_3^{n_3} b_1^{n_1} b_2^{n_2} b_3^{n_3} \mid n_1, n_2, n_3 \geqslant 1\}$$

is not expressible as the intersection of two context-free languages. Their arguments can be modified to show that L cannot be expressed as the union of a finite number of intersections of two context-free languages. This gives us our contradiction. $\square$

This language L of Theorem 6.5 is expressible as the intersection of three context-free languages, the first comparing the a_1's and b_1's and b_2's and c_2's, the second comparing the a_2's and b_2's and b_3's and c_3's, and the third comparing the a_1's and c_1's and a_3's and b_3's.

Corollary 1. W_S does not contain all languages expressible as the intersection of three context-free languages.

Corollary 2. W_S is incomparable with the intersection closure and with the Boolean closure of the context-free languages.

Proof. The language $\{a^n \mid n \geqslant 4,\ n \text{ not prime}\}$ is in W_S as shown by Example 3 but cannot be in the Boolean closure of the context-free languages. $\square$

Corollary 3. W_S is not closed under inverse homomorphism or under intersection with regular sets.

Proof. Let L be defined as in Theorem 5.5 and let L_1, L_2, L_3 be three context-free languages such that $L = L_1 \cap L_2 \cap L_3$. For each i let $L = L(G_i)$ for a context-free grammar $G_i = (V_i, \Sigma, P_i, S_i)$ with $V_i \cap V_j = \Sigma$ for $i \neq j$. Let d, σ be new and

$$P_h = \{\sigma \to \langle S_1 \rangle, \langle S_2 \rangle \to S_2 \langle S_2 \rangle d, \langle S_3 \rangle \to d\}$$

and let G be the simple W-grammar

$$G = ((V_1 - \Sigma) \cup (V_2 - \Sigma), \Sigma \cup \{\sigma, d\}, \Sigma \cup \{d\}, P_1 \cup P_2, P_h, \sigma)$$

Then $L(G) = \{w^n d^{n+1} \mid n \geqslant 0, w \in L\}$ and $Ldd = L(G) \cap \Sigma^+ dd$. If $\bar{c}_3$ is new and $h(a) = a$, for $a \in \Sigma$, $h(\bar{c}_3) = c_3 dd$, then

$$L_1 = h^{-1}(L(G)) = \{a_1{}^n a_2{}^m a_3{}^k b_1{}^n b_2{}^m b_3{}^k c_1{}^n c_2{}^m c_3^{k-1} \mid n, m, k \geqslant 1\}\bar{c}_3$$

The arguments used to show $L \notin W_S$ can also be used to show Ldd and L_1 not in W_S. □

We could continue in this vein to establish other nonclosure properties of W_S. We summarize:

Theorem 6.6. W_S is not closed under: union, concatenation, nonerasing homomorphism, inverse homomorphism, and intersection with regular sets.

Thus W_S does not possess any of the "AFL" closure properties (union, concatenation, Kleene +, nonerasing homomorphism, inverse homomorphism, and intersection with regular sets) except Kleene +. Since $\mathscr{Q}$ is an AFL and every member of $\mathscr{Q}$ is the nonerasing homomorphic image of the intersection of three context-free languages,[8] the proof of Corollary 3 of Theorem 6.5 can be used to show that $\mathscr{Q}$ is the least AFL containing W_S and each member of $\mathscr{Q}$ can be expressed as $h_1(h_2^{-1}(L))$ for $L \in W_S$ and h_1 and h_2 nonerasing homomorphisms.

In the last section we examine some of the open problems on the relationship between W_S, $\mathscr{Q}$, and some subclasses of W-grammars.

7. FURTHER QUESTIONS

The study of regular-based and simple W-grammars leads to many other questions, some of them on the precise relationship between W_S and W_{RB} and various well-known families of languages, and others regarding extensions of W-grammars.

We saw that finiteness is decidable for normal regular-based W-grammars and that members of W_{RB} are quasirealtime. It is likely that both statements apply also to macrogrammars and to indexed languages. That is, we conjecture that finiteness is decidable for macrogrammars (under either the outside-in or inside-out definitions) and that indexed languages are quasi-realtime, i.e., belong to $\mathscr{Q}$.

The exact relationship between the family $\mathscr{I}$ of indexed languages and W_{RB} is unknown. We know that $W_{RB} \subsetneq \mathscr{I}$ but do not know exactly how W_{RB} should be extended to obtain $\mathscr{I}$. One possibility is that $\mathscr{I}$ is the closure of W_{RB} under finite state translations; that does not seem too likely, however. In view of the similarities between the operations of factored, normal, regular-based W-grammars and those of macrogrammars, it seems plausible that some natural relationship between W_{RB} and $\mathscr{I}$ exists.

The relationship between W-grammars—and W_{RB} in particular—and various tree manipulating systems might be a fruitful area of research.

Several families of languages are closely related to W_S: $\mathscr{Q}$, the family of languages accepted by nondeterministic multitape Turing machines in realtime, $\mathscr{Q}_2$, the family of languages accepted by nondeterministic two pushdown store machines in realtime, W_u, the family of languages by unary W-grammars, and W_{NL}, the family of languages generated by normal lossless W-grammars. It is fairly evident that:

1. $W_S \subsetneq W_u \subseteq \mathscr{Q} \subseteq W_{NL} \subseteq \mathrm{TIME}(n^2)$, where $\mathrm{TIME}(n^2)$ is the family of languages accepted by nondeterministic multitape Turing machines in time proportional to the square of the length of the input.
2. $\mathscr{Q}_2 \subseteq W_u$.
3. $\mathscr{Q}$ is the closure of W_S under nonerasing finite-state translations.

We conjecture that $W_{NL} = \mathscr{Q}$. The difficulty lies in hyperrules such as $\langle A \rangle \to \langle uAv \rangle$ which might require checking membership of w, uwv, $uuwvv$, etc. in a context-free language and thus scanning a word w more than a fixed number of times. Thus the approach used in showing $W_{RB} \subseteq \mathscr{Q}$ and $W_S \subseteq \mathscr{Q}$ yields at best time n^2. It seems plausible that a tighter construction would yield $W_{NL} \subseteq \mathscr{Q}$. On the other hand, $W_S \subseteq \mathscr{Q}_2$ if and only if $\mathscr{Q}_2 = \mathscr{Q}$; the latter question is still open.

There are various ways in which the concept of a W-grammar or double-level grammar can be generalized. There does not seem too much point in extending unrestricted W-grammars since they already represent all recursively enumerable sets quite conveniently. However, extensions of some of the restricted families, particularly of W_{RB}, might prove useful.

For a family of languages $\mathscr{L}$ we can speak of an $\mathscr{L}$-based double-level grammar as a sixtuple $(V_M, V_P, \Sigma, P_h, \sigma, \mu)$, where V_M, V_P, Σ, P_h, and σ are as before and μ maps each A in V_M into a language $\mu(A) \in \mathscr{L}$. One might treat probabilistic regular-based double-level grammars in which distributions were attached to the regular languages $\mu(A)$, to the assignment function μ, or to the use of hyperrules—or even to all three. One might also consider double-level Lindenmeyer systems in which each $\mu(A)$ is an OL language and the derivation process required simultaneous expansion of each protovariable or hypernotion in the string (cf. Rozenberg and Doucet[(21)]).

There are two obvious ways in which the derivation process could be extended. One might allow the expansion of a protovariable *within* a hypernotion—e.g., $\langle aZb \rangle \Rightarrow \langle ay\langle bcZ \rangle b \rangle$—or allow $\mu(A)$ to contain nested hypernotions—e.g., $\langle ab\langle baa \rangle\langle a\langle a \rangle\rangle\rangle$—and then use either outside-in or inside-out derivations. Another possibility is to allow words in $\mu(A)$ to contain a fixed number of occurrences of a special comma symbol. Then if

w_1, w_2, w_3 and α, β, γ are words in $\mu(A)$, a hyperrule $\langle A\rangle \to A_1\langle A\rangle A_2\langle A_3, A_2\rangle$ would correspond to productions

$$\langle w_1, w_2, w_3\rangle \to w_1\langle w_1, w_2, w_3\rangle\, w_2\langle w_3, w_2\rangle$$

and

$$\langle \alpha, \beta, \gamma\rangle \to \alpha\langle \alpha, \beta, \gamma\rangle\, \beta\langle \gamma, \beta\rangle.$$

Some of these ideas might shed light on the relationship between W_{RB} and $\mathscr{I}$.

REFERENCES

1. A. V. Aho, "Indexed grammars—an extension of context-free grammars," *JACM* **15**:647–671 (1968).
2. A. V. Aho, "Nested stack automata," *JACM* **16**:383–406 (1969).
3. A. V. Aho and J. Ullman, *The Theory of Parsing, Translation and Compiling*, Volume I: *Parsing* (Prentice-Hall, 1972).
4. John Luther Baker, "Some formal properties of the syntax of ALGOL 68," Doctoral Dissertation, University of Washington (1970).
5. Y. Bar-Hillel, M. Perles, and E. Shamir, "On formal properties of simple-phrase structure grammars," *Z. Phonetik, Sprachwiss., Kommunikationsforsch.* **14**:143–172 (1961).
6. R. V. Book, "Grammars with time functions," Doctoral Dissertation, Harvard University (1969).
7. R. V. Book, "Time-bounded grammars and their languages," *JCSS* **5**:397–429 (1971).
8. R. V. Book and S. A. Greibach, "Quasi-realtime languages," *Mathematical Systems Theory* **4**:77–111 (1970).
9. G. de Chastellier and A. Colmerauer, "*W*-grammar," in *Proc. ACM National Conference* (1969), pp. 511–518.
10. J. C. Cleaveland and R. C. Uzgalis, "What every programmer should know about grammar," Modeling and Measurement Note No. 12, Computer Science Department, University of California, Los Angeles (1973).
11. M. J. Fischer, "Grammars with macrolike productions," Doctoral Dissertation, Harvard University (1968).
12. M. J. Fischer, "Grammars with macrolike productions," in *Proc. IEEE Ninth Annual Symposium on Switching and Automata Theory* (Schenectady, New York, October 1968), pp. 131–142.
13. S. Ginsburg, S. A. Greibach, and M. A. Harrison, "One-way stack automata," *JACM* **14**:389–418 (1967).
14. S. Ginsburg and H. G. Rice, "Two families of languages related to ALGOL," *JACM* **9**:350–371 (1962).
15. J. Hopcroft and J. D. Ullman, *Formal Languages and Their Relation to Automata* (Addison-Wesley, 1969).
16. L. Y. Liu and P. Weiner, "An infinite hierarchy of intersections of context-free languages," *Mathematical Systems Theory* **7**:185–192 (1973).
17. A. W. Mazurkiewicz, "A note on enumerable grammars," *Information and Control* **14**:555–558 (1969).
18. A. R. Meyer, P. C. Fischer, and A. L. Rosenberg, "Turing machines with several read-write heads," *Proc. IEEE Eighth Annual Symposium on Switching and Automata Theory* (1967), pp. 148–154.

19. P. Nauer (ed.), "Revised report on the algorithmic language ALGOL 60," *Comm. ACM* **6**:1–17 (1963).
20. A. Nerode, "Linear automaton transformations," *Proc. Amer. Math. Soc.* **9**:541–544 (1958).
21. G. Rozenberg and P. Doucet, "On *O-L* languages," *Information and Control* **19**:302–318 (1971).
22. M. Sintzoff, "Existence of van Wijngaarden's syntax for every recursively enumerable set," *Ann. Soc. Sci. de Bruxelles* **2**:115–118 (1967).
23. R. E. Stearns and P. M. Lewis, "Property grammars and table machines," *Information and Control* **14**:524–549 (1969).
24. A. van Wijngaarden (ed.), "Report on the algorithmic language ALGOL 68," *Numerische Mathematik* **14**:79–218 (1969).
25. Peter Wegner, "The Vienna definition language," *Computing Surveys* **4**:5–63 (1972).