COMPARATIVE COMPLEXITY OF GRAMMAR FORMS

Seymour Ginsburg and Nancy Lynch
University of Southern California
Los Angeles, California 90007

I. INTRODUCTION

The definition of "grammar form" introduced in [CG] makes it possible to state and prove results about various types of grammars in a uniform way. Among questions naturally formalizable in this framework are many about the complexity or efficiency of grammars of different kinds. For example, one might wish to know by how much it is possible to improve the efficiency of right-linear form for expressing regular sets, by using another type of grammar [MF]. Grammar forms provide a reasonable way of considering the totality of other forms we might use, and so answering the question with both upper and lower bound results.

The general question considered in this paper is the following: which grammar forms are more efficient than other grammar forms for the expression of classes of languages, and how much gain in efficiency is possible? Our results deal solely with context-free grammars, and use both derivation complexity and size of grammars as complexity measures. Our most interesting results are a general form-preserving speed-up theorem for derivation complexity, and a pair of results giving upper and lower bounds on the amount of size improvement possible over right-linear form, using forms whose expressive power is exactly the regular sets. Results related to the latter pair are given for classes of languages other than the regular sets; many open questions remain.

The basic grammar form model is given by:

Definition 1.1: A (context-free) grammar form is a 6-tuple $F = (V, \Sigma, \mathcal{V}, \mathcal{A}, \Theta, \sigma)$, where

(i) V is an infinite set of abstract symbols,
(ii) Σ is an infinite subset of V such that $V-\Sigma$ is infinite, and
(iii) $G_F = (\mathcal{V}, \mathcal{A}, \Theta, \sigma)$ is a context-free grammar with $\mathcal{A} \subseteq \Sigma$ and $(\mathcal{V}-\mathcal{A}) \subseteq (V-\Sigma)$.

An interpretation of a grammar form $F = (V, \Sigma, \mathcal{V}, \mathcal{A}, \Theta, \sigma)$ is a 5-tuple $I = (\mu, V_1, \Sigma_1, P_1, S_1)$, where

(i) μ is a substitution on $\mathcal{V}^*$ such that $\mu(a)$ is a finite subset of Σ^*, $\mu(\xi)$ is a finite subset of $V-\Sigma$ for each ξ in $\mathcal{V}-\mathcal{A}$ and $\mu(\xi) \cap \mu(\eta) = \emptyset$ for each ξ and η, $\xi \neq \eta$, in $\mathcal{V}-\mathcal{A}$,
(ii) P_1 is a subset of $\mu(\Theta) = \bigcup_{\pi \text{ in } \Theta} \mu(\pi)$, where $\mu(\alpha \rightarrow \beta) = \{u \rightarrow v \mid u \text{ in } \mu(\alpha), v \text{ in } \mu(\beta)\}$,
(iii) S_1 is in $\mu(\sigma)$, and
(iv) Σ_1 (V_1) is the set of all symbols in $\Sigma(V)$ which occur in P_1 (together with S_1).

We use the notation $\mathcal{L}(F)$ for the set of all grammars which arise from interpretations of form F, $L(G)$ for the language generated by grammar G, and $\mathcal{L}(F)$ for $\{L \mid (\exists G \in \mathcal{L}(F))[L(G) = L]\}$.

Right-linear form is the grammar form $< V, \Sigma, \{\sigma, a\}, \{a\}, \{\sigma \rightarrow a\sigma, \sigma \rightarrow a\}, \sigma >$. Left-linear form is the grammar form $< V, \Sigma, \{\sigma, a\}, \{a\}, \{\sigma \rightarrow \sigma a, \sigma \rightarrow a\}, \sigma >$. Standard linear form is the grammar form $< V, \Sigma, \{\sigma, a\}, \{a\}, \{\sigma \rightarrow a\sigma a, \sigma \rightarrow a\}, \sigma >$. Chomsky form is the grammar form $< V, \Sigma, \{\sigma, a\}, \{a\}, \{\sigma \rightarrow \sigma\sigma, \sigma \rightarrow a\}, \sigma >$.

II. DERIVATION COMPLEXITY

We first ask whether some grammar forms provide interpretations with smaller derivation complexity (as studied by Gladkii [Gl] and Book [B]) than other grammar forms, for the expression of particular context-free languages. Theorem 2.2 below answers this question in the negative; all context-free forms are seen to be equally efficient for the expression of all possible context-free languages, if the sole measure of efficiency is taken to be the lengths of derivations.

Definition 2.1: If G is a context-free grammar, Φ_G is a function defined for words in L(G), giving the fewest steps in any derivation of the word in G. $\varphi_G = \min_{x \in L(G), x \neq \epsilon} \Phi_G(x)$. ($\epsilon$ denotes the empty word.) A form F is minimal if for each language L in $\mathcal{L}(F)$ and each natural number n, there is a grammar G in $\mathcal{L}(F)$ with $L(G) = L$ and $\Phi_G(x) \leq \max\{\varphi_{G_F}, \frac{|x|}{n}\}$ for all x in L.

Thus, for a minimal form F, every language in $\mathcal{L}(F)$ may be derived in form F in a linear number of steps, with constant of linearity as small as desired. Justification for the term "minimal" comes from the fact that for any G in $\mathcal{L}(F)$, there is an n such that any non-ϵ word x in L(G) has $\Phi_G(x) \geq \max(\varphi_{G_F}, \frac{|x|}{n})$.

The main result of this section is:

Theorem 2.2: Every form is minimal.

Proof: The most interesting forms for classical language theory are forms F with $\mathcal{L}(F)$ = the regular sets, $\mathcal{L}(F)$ = the context-free languages and $\mathcal{L}(F)$ = the linear languages; each of these cases requires an individual proof. For arbitrary forms, with arbitrary language classes, we first show that transformation to a "sequential normal form" does not affect the minimality property. Then we induct on the sequential structure of the

form (the basis being simply the interesting cases proved separately). We outline the main ideas b below; the complete proof appears in [GL1].

A grammar form F is vacuous if $L(G_F) = \emptyset$ or $L(G_F) = \{\epsilon\}$. For nonvacuous forms, a "finite patch" may be made for short words:

Lemma 2.3: Let F be a nonvacuous form and n a positive integer. Suppose there exists a positive integer k and a grammar G in $\mathcal{L}(F)$ such that $\phi_G(x) \leq \max\{k, \frac{|x|}{n}\}$ for all x in L(G). Then there exists a grammar G′ in $\mathcal{L}(F)$ such that $L(G') = L(G)$ and $\phi_{G'}(x) \leq \max\{\varphi_{G_F}, \frac{|x|}{n}\}$ for all x in L(G′).

Proof of Lemma 2.3: We add to G productions that generate, in φ_{G_F} steps, the finite number of words x in L(G) for which $\frac{|x|}{n} < k$. □

We next show that minimality is not affected by either adding or removing "redundant" productions from a grammar form:

Lemma 2.4: Let F be a nonvacuous form and $\beta \overset{*}{\underset{G_F}{\Longrightarrow}} w$ a derivation, with β a variable. Let F′ be the grammar form obtained by adding to F the production β → w. Then F is minimal if and only if F′ is minimal.

Proof of Lemma 2.4: By Lemma 2.3 and Proposition 2.1 of [CG], F minimal implies F′ is minimal. If F′ is minimal, and we want a language to be expressed in $\mathcal{L}(F)$ with constant of linearity n, we first express it in $\mathcal{L}(F')$ with constant of linearity kn, where k is the number of steps it takes to simulate β → w in F. A natural translation then gives the needed grammar in $\mathcal{L}(F)$. □

Lemmas 2.3 and 2.4 are used repeatedly in proofs throughout this section.

Lemma 2.5: In each of the following cases, F is minimal:
(a) $\mathcal{L}(F)$ is the family of all finite sets.
(b) $\mathcal{L}(F)$ is the family of all regular sets.
(c) $\mathcal{L}(F)$ is the family of all linear languages.
(d) $\mathcal{L}(F)$ is the family of all context-free languages.

Proof of Lemma 2.5: (a) Since $\mathcal{L}(F)$ is the family of all finite sets, there is a terminal word $w \neq \epsilon$ with $\sigma \overset{*}{\underset{G_F}{\Longrightarrow}} w$. Let F′ be the form obtained from F by adding the production σ → w. By Lemma 2.4, it suffices to show F′ is minimal. But a trivial grammar for any finite set demonstrates this fact.

(b) Since $\mathcal{L}(F)$ is the class of all regular sets, $L(G_F)$ is an infinite set. Then there exist x_1, x_2, x_3, x_4, x_5 in $\mathcal{A}^*$ and β in $\mathcal{V}$-$\mathcal{A}$ such that $x_5 \neq \epsilon$, $x_3x_4 \neq \epsilon$, $\sigma \overset{*}{\underset{G_F}{\Longrightarrow}} x_1\beta x_2$, $\beta \overset{*}{\underset{G_F}{\Longrightarrow}} x_3\beta x_4$ and $\beta \overset{*}{\underset{G_F}{\Longrightarrow}} x_5$. By 2.4, there is no loss of generality in assuming $\sigma \to x_1\beta x_2$, $\beta \to x_3\beta x_4$ and $\beta \to x_5$ are in G_F (so that G_F is essentially like right-linear or left-linear form). Assume $x_3 \neq \epsilon$, by symmetry.

Let L be any regular set. Then L = L(G) for some right-linear grammar G, with start symbol S, having no productions of the type A → B. If n is any positive integer, define a new grammar G′, with start symbol S′, having the productions

$S' \to S$,

$A \to wB$ if $A \overset{\leq n+1}{\underset{G}{\Longrightarrow}} wB$,

and

$A \to w$ if $A \overset{\leq n+1}{\underset{G}{\Longrightarrow}} w$.

(The symbol above the arrow indicates the number of steps in the derivation.) G "solves" several productions of G in a single production, thus speeding up the generation of words.

(c) Similar to (b).

(d) Cases (b) and (c) presented little difficulty, as we can "solve" as many productions as we like of a right-linear, left-linear or standard linear grammar, to obtain other productions of the same type. In case (d), we cannot simply follow the same procedure, since the resulting productions will not necessarily be of the required form. Instead, we introduce productions where possible which simulate within Chomsky form the result of composing sequences of productions:

Since $\mathcal{L}(F)$ is the family of all context-free languages, there exist $x_1, x_2, x_3, x_4, x_5, x_6$ in $\mathcal{A}^*$, $x_6 \neq \epsilon$, β in $\mathcal{V}$-$\mathcal{A}$ with $\sigma \overset{*}{\underset{G_F}{\Longrightarrow}} x_1\beta x_2$, $\beta \overset{*}{\underset{G_F}{\Longrightarrow}} x_3\beta x_4\beta x_5$, and $\beta \overset{*}{\underset{G_F}{\Longrightarrow}} x_6$. We assume F contains these productions explicitly, as before.

Let L be any context-free language, G a grammar for L in Chomsky form, where S, the start symbol, never occurs on the right of a production, and where only S can generate ε. Let n be any positive integer. We define a grammar G′ with start symbol S′, and the following productions:

$S' \to S$,

$A \to w$, if $A \overset{\leq 112n}{\underset{G}{\Longrightarrow}} w$,

$A \to BC$, if $A \to BC$ is in G,

and

$A \to DE$, $E \to BH$, $D \to v$, $H \to w$, if $A \overset{\leq 112n}{\underset{G}{\Longrightarrow}} vBw$.

G′ will be the required grammar providing n as constant of linearity. G′ speeds up G's derivations in two ways. First, G′ deposits short words in a single step. Second, if G causes a variable to generate a single variable with short words on both sides, G′ does this in a fixed number of steps. We give a brief argument to show that G′ causes at least n terminal symbols to be deposited at each step, on the average:

Let x be an arbitrary word in L. For each G′-derivation of x, there is associated a G′-derivation tree. We consider the tree with the fewest nodes, delete the root, all leaf nodes and all edges incident to these nodes. This gives us a binary tree T(x) describing a shortest G′-derivation for x. By the definition of G′ and the choice of T(x), this tree may be shown to have the following three properties:

(1) No two consecutive internal nodes of T(x) have both their node names new variables of G′ (i.e., variables not in G),

(2) Each internal node of T(x) with at least one son an internal node generates a subtree of T(x) whose terminal word is of length ≥ 56n, and

(3) Let $n_1, \ldots, n_6$ be six consecutive internal nodes of T(x), and for each i, let m_i be the other son of n_{i-1}. If each m_i is a leaf in T(x) and if

$B_i \to w_i$ in $T(x)$ (B_i the node name of m_i), then $|w_2 \ldots w_s| \geq 56n$.

That is, any internal node generates a long word relative to n, and any group of consecutive internal nodes with one son a leaf generates a variable and a long word relative to n.

A graph-theoretic argument now shows that the total number of symbols generated by $T(x)$ is at least n times the number of nodes in $T(x)$, i.e., that $\Phi_{G'}(x) \geq \frac{|x|}{n}$, as needed. □

We next move from forms for specific language classes to general grammar forms. We introduce a normal form:

Lemma 2.6: For every form F, there exists a form F with $\mathcal{L}(F') = \mathcal{L}(F)$, F' completely reduced and sequential, and F minimal if F' is minimal. (A context-free grammar is sequential if the variables can be ordered $S = A_1, \ldots, A_r$ such that if $A_i \to uA_jv$ is any production, then $j \geq i$. A form F is sequential if G_F is sequential.)

Proof of Lemma 2.6: If F is vacuous, the result is trivial. Otherwise, we may follow the transformation procedure in [CG], Section 3. We note that no step of that procedure changes a nonminimal form into a minimal one. The argument at each step is similar to that for Lemma 2.4. □

Proof of Theorem 2.2, continued: Every vacuous form is minimal, so we restrict attention to nonvacuous forms. By Lemma 2.6, we may further restrict attention to completely reduced, sequential forms. The proof is by induction on the number k of variables in the grammar form F.

If $k = 1$, Lemma 5.1 of [CG] shows that $\mathcal{L}(F)$ is either the finite, regular, linear or context-free languages. By Lemma 2.5, F is minimal.

Assume the theorem is true for all completely reduced, sequential grammar forms with at most k variables, and let F have k+1 variables. The variables of F can be arranged into a sequence $\sigma = \alpha_0, \alpha_1, \ldots, \alpha_k$ in sequential order, where σ is the start variable of F. By Lemma 2.5, we may assume $\mathcal{L}(F)$ is not the family of all context-free languages, and that $\mathcal{L}(G_F)$ is infinite. We may also assume:

(1) Every variable of F generates a nonempty terminal word,

(2) If $\sigma \overset{*}{\underset{G_F}{\Longrightarrow}} w_1\sigma w_2$ for any words w_1, w_2, then there are terminal words x_1, x_2 with $\sigma \to x_1\sigma x_2$ in G_F,

(3) If $\sigma \overset{*}{\underset{G_F}{\Longrightarrow}} w\sigma$ (or $\sigma \overset{*}{\underset{G_F}{\Longrightarrow}} \sigma w$) for any word w, then there is a terminal word x with $\sigma \to x\sigma$ (or $\sigma \to \sigma x$) in G_F.

For each $i \geq 1$, let F_i be the grammar form with start variable α_i which contains all productions of F involving only variables generable from α_i. (We will later apply the inductive hypothesis to these forms.)

Let L be in $\mathcal{L}(F)$, n any positive integer, and G any grammar in $\mathcal{L}(F)$ with $L(G) = L$. We modify G to obtain a new grammar G' yielding the speedup of L by constant n. Let A be a variable of G in $\mu(\alpha_i)$ for some $i \geq 1$. Let G_A be the grammar with start symbol A which contains exactly those productions of G which are interpretations of productions in F_i. $G_A \in \mathcal{L}(F_i)$, so by induction, there is a grammar $G'_A \in \mathcal{L}(F_i)$ with $L(G'_A) = L(G_A)$ and $\Phi_{G'_A}(x) \leq \max\{\varphi_{G_{F_i}}, \frac{|x|}{2(n+1)}\}$ for all x. We can assume A is the start symbol of G'_A.

Let s be the number of elements in $\mu(\sigma)$, r the maximum number of (not necessarily distinct) variables on the right side of any production of F. G' will contain the following productions:

All productions of G,

$A \to w$, if w is a terminal string, $|w| \leq 2(n+1)(r+2)$ and $A \overset{*}{\underset{G}{\Longrightarrow}} w$,

$A \to x_1Bx_2$, if $A, B \in \mu(\sigma)$, $A \overset{\leq (n+1)s}{\underset{G}{\Longrightarrow}} w_1Bw_2$ for some words w_1, w_2, if $B_1, \ldots, B_q$ are the (not necessarily distinct) variables of G appearing in w_1w_2 in order from left to right, if for each i, $B_i \overset{*}{\underset{G}{\Longrightarrow}} u_i$, u_i a terminal word with $|u_i| \leq 2(n+1)(r+2)$, and if x_1 and x_2 are the words obtained by replacing each B_i by u_i in w_1 and w_2, respectively,

All productions of G'_A for each $A \in \mu(\alpha_i)$, $i \geq 1$.

The three new types of productions in G' have the following purposes: the first type speeds up the derivation of short terminal words from variables. The second type speeds up derivations involving variables in $\mu(\sigma)$ and short terminal words. The third type speeds up derivations of long words from variables not in $\mu(\sigma)$.

By considering the possible forms a derivation in G can take, we may verify that G' speeds up all parts of the derivation, and thus provides the needed linear speedup. □

We note that Book [B] proves a result of a speedup flavor similar to Theorem 2.2. His speedup, however, does not preserve form; in fact, context-free grammars are sped up by grammars which are not context-free. The idea of preservation of the form of a grammar is a new one.

Theorem 2.2 has shown that all context-free forms are equally efficient, if only lengths of derivations are of interest. However, the cost of this equality is a possible exponential increase in the number of productions needed. This increase leads us to consider size of grammars as a second basis for comparison of forms.

III. SIZE COMPLEXITY

Our results about size of grammars use the following four measures of size (measures are the same as used by Gruska [Gr]).

Definition 3.1: If G is a context-free grammar, S(G) is the total number of (not necessarily distinct) variable and terminal symbols on both sides of all productions of G. (Any occurrence of the symbol ϵ is not counted.) V(G) is the total number of (not necessarily distinct) variable symbols on both sides of all productions in G. P(G) is the number of productions of G. N(G) is the number of distinct variables of G.

We prove two results about forms for the regular sets. Specifically, we characterize the amount of improvement possible over right-linear (or left-linear) form. An easy proposition, proved by a reversal construction which is fundamental to later proofs, is that right- and left-linear forms are of equal efficiency.

Proposition 3.2: Assume G is any grammar in right-linear (left-linear) form. Then there exists G' in left-linear (right-linear) form with

$L(G') = L(G)$ and

$$S(G') \leq S(G) + P(G) + 1,$$
$$V(G') \leq V(G) + P(G) + 1,$$
$$P(G') \leq P(G) + 1, \text{ and}$$
$$N(G') \leq N(G) + 1.$$

Proof: If G is in right-linear form, with start symbol S, we obtain G', with S' (a new symbol) as its start symbol. If a production of the type $A \rightarrow wB$ is in G, we put the corresponding production $B \rightarrow Aw$ in G'. If a production $A \rightarrow w$ is in G, we put $S' \rightarrow Aw$ into G'. We also put into G' the production $S_1 \rightarrow \epsilon$.

It should be clear that G' simulates the action of G "in reverse." □

We now consider arbitrary forms for the regular sets, to determine if any are substantially more efficient than right- or left-linear form, for the representation of particular regular sets. We begin by showing that every form for the regular sets has a polynomial that bounds its improvement over right- or left-linear form:

Theorem 3.3: If F is any form for the regular sets, there exist constants c and n with the following property: for every $G \in \mathcal{L}(F)$, there exists G' in right-linear form with $L(G') = L(G)$,

$$S(G') \leq c[S(G)]^n,$$
$$V(G') \leq c[V(G)]^n,$$
$$P(G') \leq c[P(G)]^n, \text{ and}$$
$$N(G') \leq c[N(G)]^n.$$

Proof: We outline a proof which gives the theorem for the first three measures only. (Bounding N(G') as above is more complicated.) The complete version of this and other results in this section will appear in [GL2].

As for Theorem 2.2, we pass first to a normal form:

Lemma 3.4: Let F be any form. There exist a form F' and constants c, n with the following properties:

(i) $\mathcal{L}(F') = \mathcal{L}(F)$

(ii) F' is completely reduced and sequential, and

(iii) if $G \in \mathcal{L}(F)$ then there exists $G' \in \mathcal{L}(F')$ with $L(G') = L(G)$, $S(G') \leq c[S(G)]^n$, $V(G') \leq c[V(G)]^n$, $P(G') \leq c[P(G)]^n$, and $N(G') \leq N(G)$.

Proof of Lemma 3.4: As in the proof of Lemma 2.6, we follow the constructions in Section 3 of [CG]. This time, we show that the growth in size of interpretation grammars is appropriately bounded at each step. □

Next, we make the right-hand sides of productions "binary:"

Lemma 3.5: Let F be any form for the regular sets. There exist a form F' for the regular sets and constants c, n with the following properties:

(i) Every production of F' is of one of the types $\alpha \rightarrow \beta\gamma$, $\alpha \rightarrow w\beta$, $\alpha \rightarrow \beta w$ or $\alpha \rightarrow w$, where α, β, γ are variables and w is a terminal string,

(ii) F' is sequential, reduced, and every variable generates a nonempty terminal string, and

(iii) if $G \in \mathcal{L}(F)$, there exists $G' \in \mathcal{L}(F')$ with $L(G') = L(G)$ and $S(G') \leq c[S(G)]^n$, $V(G') \leq c[V(G)]^n$, and $P(G') \leq c[P(G)]^n$.

Proof of Lemma 3.5: Given F, we obtain F'' from Lemma 3.4. We than transform F'' so that productions are of the type required in (i) above. To do this, productions with long right-hand sides are simulated by a series of productions of the allowable types. Although this transformation destroys the "completely reduced" property of F'', it can be done in such a way as to preserve the properties required in (ii) (provided we are careful about choosing the proper end of a long right-hand side at which to begin our simulation.) □

Lemma 3.5 shows that it is sufficient to restrict attention to normal forms, in showing that any form for the regular sets is simulable by right-linear (and therefore by left-linear) form with at most polynomial loss of efficiency. For any form of the type described in Lemma 3.5, each variable may embed itself on the right or the left, but not both. We next show how to transform such a form into one in which variables may only embed themselves on the right. The technique is similar to the reversal technique used to prove Proposition 3.1. We define a sequence $\{F_n\}$ of forms for the regular sets, of successively greater "sequential depth," in which every variable embeds itself on the right only:

Definition 3.6: For any $n \geq 1$, let F_n denote the form whose variables are $\sigma = \alpha_0$ (the start symbol), $\alpha_1, \ldots, \alpha_{n-1}$, whose only terminal symbol is a, and whose productions are:

$\alpha_i \rightarrow \alpha_j\alpha_k$, $i \leq k$, $i < j$
$\alpha_i \rightarrow a\alpha_j$, $i \leq j$,
$\alpha_i \rightarrow \alpha_j a$, $i < j$, and
$\alpha_i \rightarrow a$, for all i.

(In particular, F_1 is right-linear form.)

Lemma 3.7: Assume F is a form for the regular sets, satisfying:

(i) Every production of F is of one of the types $\alpha \rightarrow \beta\gamma$, $\alpha \rightarrow w\beta$, $\alpha \rightarrow \beta w$ or $\alpha \rightarrow w$, where α, β, γ are variables and w is a terminal string, and

(ii) F is sequential, reduced, and every variable generates a nonempty terminal string.

Then for some n, every G in $\mathcal{L}(F)$ has a G' in $\mathcal{L}(F_n)$ with $L(G') = L(G)$, $S(G') \leq 5[S(G)]^2$, $V(G') \leq 5[V(G)]^2$, $P(G') \leq 4[P(G)]^2$, and $N(G') \leq 4[N(G)]^3$.

Proof of Lemma 3.7: We assume the variables of F are $\sigma = \alpha_0$ (the start variable), $\alpha_1, \ldots, \alpha_{n-1}$ for some $n \geq 1$. This n will be the n of the lemma. We consider any $G \in \mathcal{L}(F)$, assumed without loss of generality to be reduced. Productions of G are of the following types:

(1) $A \rightarrow BC$, where $A, B \in \mu(\alpha_i), C \in \mu(\alpha_j)$, $i < j$,
(2) $A \rightarrow BC$, where $A \in \mu(\alpha_i), B \in \mu(\alpha_j), C \in \mu(\alpha_k)$, $i < j$, $i \leq k$,
(3) $A \rightarrow wB$, where $A \in \mu(\alpha_i), B \in \mu(\alpha_j)$, $i \leq j$,
(4) $A \rightarrow Bw$, where $A, B \in \mu(\alpha_i)$, for any i,
(5) $A \rightarrow Bw$, where $A \in \mu(\alpha_i), B \in \mu(\alpha_j)$, $i < j$,
and (6) $A \rightarrow w$, where A is any variable.

G' will contain all productions of G of types (2), (3), (5) and (6). We will simulate the effects of groups of productions of types (1) and (4) in reverse (as for Proposition 3.1). To do this,

we put into G′ the following seven sets of productions. (A string of terminals and high variables in the sequential order is being generated, with the productions in (a)-(d) beginning the generation at the leftmost end of the string, the productions in (e) and (f) continuing the process from left to right, and (g) concluding the generation.)

(a) $A \rightarrow w[BA]$ if $A, B \in \mu(\alpha_i)$ and $B \rightarrow w$ is in G,

(b) $A \rightarrow C[BCA]_1$
$[BCA]_1 \rightarrow w[BA]$ if $A, B \in \mu(\alpha_i), C \in \mu(\alpha_j)$, $i < j$ and $B \rightarrow Cw$ is in G,

(c) $A \rightarrow w[BCA]_2$
$[BCA]_2 \rightarrow C[BA]$, if $A, B \in \mu(\alpha_i), C \in \mu(\alpha_j)$, $i < j$, and $B \rightarrow wC$ is in G,

(d) $A \rightarrow C[BCA]_1$
$[BCA]_1 \rightarrow D[BA]$, if $A, B \in \mu(\alpha_i), C \in \mu(\alpha_j), D \in \mu(\alpha_k)$, $i < j$, $i < k$ and $B \rightarrow CD$ is in G,

(e) $[CA] \rightarrow D[BA]$, if $A, B, C \in \mu(\alpha_i), D \in \mu(\alpha_j), i < j$ and $B \rightarrow CD$ is in G,

(f) $[CA] \rightarrow w[BA]$, if $A, B, C \in \mu(\alpha_i)$ and $B \rightarrow Cw$ is in G,

and

(g) $[AA] \rightarrow \varepsilon$, for all variables of A of G.

$G' \in \mathcal{L}(F_n)$.

Intuitively, if a variable [BA] is generated in the course of a G′ derivation, then a string x of symbols such that $B \overset{*}{\Longrightarrow}_G x$ has already been deposited to the left of [BA]; this string x consists entirely of terminal symbols and of variable symbols of G which correspond to (sequentially) higher variables of F than does A. We are simulating a part of a derivation in G proceeding from variable A, so we are waiting to discover that A generates B (along with some other symbols, perhaps) in G. In particular, when a variable [AA] is generated, we have already deposited a string x derivable from A in G; since we are simulating part of a derivation in G proceeding from variable A, we may simply erase the variable [AA]. Thus, the paired variables serve to simulate from left to right the derivation of a string derived in G from right to left.

The new triple variables are simply auxiliary variables to insure that no more than two symbols appear on the right.

Formal verification of the equality of L(G′) and L(G) is deferred to [GL2].

It remains to show:

Lemma 3.8: For any n, there exists a constant c with the following property: for every $G \in \mathcal{L}(F_n)$, there exists $G' \in \mathcal{L}(F_1)$ with $L(G') = L(G)$,
$S(G') \leq c[S(G)]^{2n}$,
$V(G') \leq c[V(G)]^{2n}$,
$P(G') \leq c[P(G)]^{2n}$,
and $N(G') \leq c[N(G)]^{2n}$.

Proof of Lemma 3.8: Given $G \in \mathcal{L}(F_n)$, we define $G' \in \mathcal{L}(F_1)$ which simulates leftmost derivations of G, and whose variables are bounded strings of variables of G. Techniques are similar to those used by Chomsky [C]. □

By combining Lemma 3.4, 3.5, 3.7, and 3.8, we obtain Theorem 3.3 for the measures S(G), V(G), and P(G). If we wish to include the measure N(G) as well, we will bypass Lemma 3.5, proceeding immediately to much more complicated versions of Lemmas 3.7 and 3.8. □

We note that, because of Lemma 3.8, the exponent n in Theorem 3.3 is closely related to the "depth of nesting" of the sequential structure of the form.

Next, we show that any polynomial improvement may actually be achieved by some form for the regular sets, at least on an infinite set of languages.

Theorem 3.9: For any positive integer n, there exists a form F for the regular sets and a constant c with the following property:
For any $k \geq 1$, there is a grammar $G \in \mathcal{L}(F)$ such that:
(1) $S(G) \leq ck$,
and (2) each G′ in right-linear form with $L(G') = L(G)$ has $N(G') \geq k^n$.
(Since S is the largest and N the smallest of the four measures, Theorem 3.9 applies to all four measures.)

Proof: We define the relevant languages $L_{n,k} = 0^*(1\,0^*)^{k^n}$. □

Lemma 3.10: For any n, there is a constant c with the following property: for any k, there is a grammar $G \in \mathcal{L}(F_n)$ with $L(G) = L_{n,k}$ and $S(G) \leq ck$.

Proof of Lemma 3.10: Straightforward using the nesting capabilities of form F_n. □

Lemma 3.11: If G is a grammar in right-linear form, and $L(G) = 0^*(1\,0^*)^k$ for some positive integer k, then $N(G) \geq k+1$.

Proof of Lemma 3.11: Straightforward. □

But then Lemmas 3.10 and 3.11 combine to yield the theorem. □

Theorem 3.9 states that any polynomial improvement over right-linear form is attainable. As a corollary to Theorem 3.9 and 3.3, we see that any form for the regular sets may be similarly improved by any polynomial:

Corollary 3.12: For any form F′ for the regular sets, and any positive integer n, there exist a form F for the regular sets and a constant c with the following property:
For any $k \geq 1$, there is a grammar $G \in \mathcal{L}(F)$ such that:
(1) $S(G) \leq ck$,
and (2) each G′ in $\mathcal{L}(F')$ with $L(G') = L(G)$ has $N(G') \geq k^n$.

Thus, there is no "best" form for the regular sets.

We have thus far characterized the variations in size complexity of forms expressing exactly the regular sets. If we allow ourselves to consider forms with greater expressive power, greater improvement is possible:

Proposition 3.13: For any recursive function f, and for arbitrarily large positive integers k, there is a grammar G in Chomsky normal form, with L(G) regular,
(1) $S(G) \leq k$,
and (2) each G′ in right-linear form with $L(G') = L(G)$ has $N(G') \geq f(k)$.

Proof: This is an easy consequence of Proposition 7 of [MF], the bounded simulation of right-linear grammars by one-way finite automata, and the bounded simulation of any context-free grammar by one in Chomsky normal form. □

On the other hand, it should be noted that the improvement given by Theorem 3.9 and Proposition 3.13 is on an infinite class of regular sets,

not on all regular sets. This is necessarily the case, since we can show that there are some regular sets for which right-linear form is essentially optimal:

Theorem 3.14: For any positive integer k, there is a grammar G in right-linear form with:
(1) $V(G) \leq 7k$,
and (2) each context-free grammar G' with $L(G') = L(G)$ has $N(G') \geq k$.

Proof: Given k, consider the language

$$(0\,1)^*(0\,0\,1)^*\ldots(0^{2k-1}\,1)^*(0^{2k}\,1)^*.$$

G will be the grammar with start symbol S and productions

$S \rightarrow A_1A_2\ldots A_k$,
$A_i \rightarrow 0^{2i-1}\,1\,A_i$, $\quad 1 \leq i \leq k$,
$A_i \rightarrow A_i\,0^{2i}\,1$, $\quad 1 \leq i \leq k$,
and $A_i \rightarrow \varepsilon$ $\quad 1 \leq i \leq k$.

Now consider any G' with $L(G') = L(G)$. We claim that for any i, $1 \leq i \leq 2k$, there is a variable A_i in grammar G' such that

$$A_i \overset{*}{\underset{G'}{\Longrightarrow}} w_1A_iw_2\,1\,0^i\,1\,w_3 \text{ or } A_i \overset{*}{\underset{G'}{\Longrightarrow}} w_1\,1\,0^i\,1\,w_2A_i\,w_3,$$

where w_1, w_2, w_3 are terminal words. This is necessary to generate all the words in the language (in particular, those with many occurrences of $1\,0^i\,1$).

But then it is not difficult to show that for no three distinct i_1, i_2, i_3 can we have $A_{i_1} = A_{i_2} = A_{i_3}$. For if this situation were to occur, a "wrong word" would be generated by the grammar. Thus, there are at least k distinct variables in G'. □

We next turn to the size complexity situation for forms whose expressive power is exactly the linear languages, or exactly the context-free languages. For the linear languages, results parallel those for the regular sets:

Theorem 3.15: If F is any form for the linear languages, there exist constants c and n with the following property:
For every G in $\mathcal{L}(F)$, there exists G' in standard linear form with $L(G') = L(G)$,
$S(G') \leq c[S(G)]^n$,
$V(G') \leq c[V(G)]^n$,
$P(G') \leq c[P(G)]^n$,
and $N(G') \leq c[N(G)]^n$.

Proof: Similar to Theorem 3.3, but with many more complications in detail. The full construction for the first three measures appears in [GL2]. For the measure N(G), similar remarks to those in the proof of Theorem 3.3 apply. □

Theorem 3.16: For any positive integer n, there exists a form F for the linear languages and a constant c with the following property:
For any $k \geq 1$, there is a grammar $G \in \mathcal{L}(F)$ such that:
(1) $S(G) \leq ck$,
and (2) each G' in standard linear form with $L(G') = L(G)$ has $N(G') \geq k^n$.

Proof: The languages used are the same as in Theorem 3.9 and the proof is very similar. □

Corollary 3.12 also has an obvious analog for the linear languages.

For the case of forms expressing all context-free languages, our results collapse:

Proposition 3.17: If F is any form for the context-free languages, there exists a constant c with the following property:
For every G in $\mathcal{L}(F)$ there exists G' in Chomsky for with $L(G') = L(G)$, $S(G') \leq c[S(G)]$,
$V(G') \leq c[V(G)]$,
and $P(G') \leq c[P(G)]$.

Proof: By straightforward simulation.

IV. FURTHER STUDY

We noted at the end of Section II the apparent existence of a tradeoff between derivation complexity and size complexity. Such a tradeoff remains to be quantified and studied. Perhaps a combination of both measures is a reasonable criterion for judging efficiency of forms.

We would like to know whether results similar to those in Section III hold for other subcontext-free language classes besides the regular and linear languages. One possible difficulty is that we have as yet no "canonical forms" analogous to right-linear or standard linear form, for other language classes. However, perhaps it may be shown that any two forms with the same expressive power can simulate each other with at most polynomial loss of efficiency.

Even for regular sets, we do not know if there exist two forms, each expressing exactly the regular sets, and each of which is more efficient than the other (in size complexity) for some languages.

Grammars which are not context-free remain to be examined.

REFERENCES

[B] Book, R. V., "Time-Bounded Grammars and their Languages," JCSS, 5 (1971), pp. 397-429.

[C] Chomsky, N., "On Certain Formal Properties of Grammars," Inf. and Control, 2 (1959), pp. 137-167.

[CG] Cremers, A. B and S. Ginsburg, "Context-Free Grammar Forms," to appear in JCSS.

[GL1] Ginsburg, S. and N. A. Lynch, "Derivation Complexity in Context-Free Grammar Forms," submitted for publication.

[GL2] Ginsburg, S. and N. A. Lynch, "Size Complexity in Context-Free Grammar Forms," in preparation.

[Gl] Gladkii, A., "On the Complexity of Derivations in Phrase-Structure Grammars," Algebra i Logika Sem. 3 (1964), pp. 29-44.

[Gr] Gruska, J., "On the Size of Context-Free Grammars," Kybernetica 8 (1972), pp. 213-218.

[MF] Meyer, A. R. and M. J. Fischer, "Economy of Description by Automata, Grammars and Formal Systems," 12-th Annual Symposium on Switching and Automata Theory (1971), pp. 188-191.