A survey of grammar forms — 1977*

By S. Ginsburg

Introduction

In [3] the notion of a grammar form was abstracted to consider the situation when a master grammar[1] is given and one wishes to discuss grammars which "look like" the master one. Since then, research into grammar forms has continued at a rapid pace.[2] Moreover, other researchers have picked up on the form notion and have written extensively on L-forms (grammar forms applied to L-systems), e.g. [L1—L10]. In the present talk, I shall restrict myself almost entirely to *grammar* forms, and give a brief overview of those portions with which I am most familiar.

Throughout, I assume a general knowledge of language theory.

§1. Preliminaries

By way of motivation for "looks like" in grammar forms, consider the three context-free rules:

(1) $\xi \rightarrow a_1 \alpha a_2 \beta$,

(2) $\xi' \rightarrow w_1 \alpha' w_2 \beta'$, and

(3) $\xi' \rightarrow w_1 \alpha' w_2 \beta' w_3$,

where the Greek letters are nonterminals, the a_i are terminal symbols, and the w_j are terminal words. From an intuitive point of view, would you agree that rule 2 looks like rule 1 (because the primed nonterminals correspond to the unprimed

* The contents of this survey are a combination of two distinct talks. The first was at the Colloquium on Automata and Formal Languages, in Szeged, Hungary, August 30—September 2, 1977. The second was at the 6th International Symposium on Mathematical Foundations of Computer Science, in Tatranska Lomnica, Czechoslovakia, September 5—9, 1977.

[1] Unless otherwise stated, grammar is to mean context-free grammar.

[2] The pace can be determined by comparing the present survey with that given $2\frac{1}{2}$ years ago in [5].

nonterminals, and the terminal words correspond to the terminal symbols)? Would you also agree that (3) does *not* look like (1) (because while β' corresponds to β, w_3 does not correspond to anything to the right of β)? If your answers were yes to both questions, then you should have no difficulty in agreeing with the reasonableness of the abstraction of when one grammar looks like another.

We now formalize our ideas.

Definition. A *grammar form* is a grammar[3] $G=(V, \Sigma, P, \sigma)$, together with underlying infinite alphabets V_∞ and Σ_∞, such that $\Sigma_\infty \subseteqq V_\infty$, $V_\infty - \Sigma_\infty$ is infinite, $\Sigma \subseteqq \Sigma_\infty$, and $V - \Sigma \subseteqq V_\infty - \Sigma_\infty$.

The underlying alphabets V_∞ and Σ_∞ will always be understood. Hence we shall usually omit them and identify a grammar form with a grammar. The term "grammar form" will be employed when we wish to emphasize that the grammar G is conceived as a master grammar for describing a family of grammars, each of which looks like G. The term "grammar" will be used to indicate that the grammar G is to be considered primarily as a device generating a set of strings, i.e., generating a language.

We now specify when one grammar is to "look like" another. The mechanism for accomplishing this is an "interpretation".

Definition. An *interpretation* of a grammar form $G=(V, \Sigma, P, \sigma)$ is a 5-tuple $I=(\mu, V_I, \Sigma_I, P_I, S_I)$, where μ is a substitution on V^* satisfying

(1) $\mu(a)$ is a finite subset of Σ_∞^* for each element a in Σ, $\mu(\xi)$ is a finite subset of $V_\infty - \Sigma_\infty$ for each ξ in $V - \Sigma$, and $\mu(\alpha) \cap \mu(\beta) = \emptyset$ for all $\alpha \neq \beta$ in $V - \Sigma$.

(2) $P_I \subseteqq \bigcup_{\pi \text{ in } P} \mu(\pi)$, where $\mu(\xi \to w) = \{\alpha \to y / \alpha \text{ in } \mu(\xi),\ y \text{ in } \mu(w)\}$.

(3) S_I is in $\mu(\sigma)$.

(4) $V_I(\Sigma_I)$ contains the set of all symbols (terminals) occurring in the rules of P_I.

$G_I = (V_I, \Sigma_I, P_I, S_I)$ is called the *grammar of the interpretation*.

The grammar G_I is context free and is supposed to look like the master grammar G. The substitution μ indicates what symbols in the original grammar can be replaced by what strings, i.e., which words look like what symbols. In particular, terminals are to be replaced by strings of terminals, but nonterminals are only to be replaced by nonterminals. The condition $\mu(\alpha) \cap \mu(\beta) = \emptyset$ for all $\alpha \neq \beta$ in $V - \Sigma$ means that replacement of distinct variables must be by distinct variables. Condition 2 asserts that each rule in P_I must resemble some rule in P. Note that we do not require all rules looking like those in P to appear in G_I. Condition 3 merely says that the start variables must correspond. Condition 4 is strictly technical and asserts that the terminals in G_I come from the universal variable alphabet $V_\infty - \Sigma_\infty$.

Notation. For each grammar form G let $\mathscr{G}(G) = \{G_I / I \text{ an interpretation of } G\}$ and let $\mathscr{L}(G) = \{L(G_I) / G_I \text{ in } \mathscr{G}(G)\}$. $\mathscr{L}(G)$ is called the *grammatical family of G*.

Thus the grammar form G acts as a master grammar for all grammars in $\mathscr{G}(G)$.

[3] We assume the reader is familiar, to some extent, with context-free grammars. Here Σ is the finite set of terminals, V is the finite set of both terminals and nonterminals, P is the finite set of rules each of the shape $\xi \to w$, where ξ is a nonterminal and w is in V^*, and σ is in $V - \Sigma$.

We now illustrate the above concepts with some specific grammar forms G. The resulting $\mathcal{G}(G)$ and $\mathcal{L}(G)$ will turn out to be well-known families of grammars and languages.

Examples. (*a*) Let $G=(\{\sigma, a\}, \{a\}, P, \sigma)$, where $P=\{\sigma\to a\sigma, \sigma\to a\}$. Each rule resembling $\sigma\to a\sigma$ is of the kind $\xi\to wv$, where ξ, v are variables and w is a terminal word. The rule $\sigma\to a$ gives rise to rules $\xi\to w$, where w is a terminal word. Then $\mathcal{G}(G)$ is the family of all right-linear grammars and $\mathcal{L}(G)$ is the family of regular sets.

(*b*) Let $G=(\{\sigma, a, b, c\}, \{a, b, c\}, P, \sigma)$, with $P=\{\sigma\to a\sigma b, \sigma\to c\}$. Then $\mathcal{G}(G)$'is the family of all linear grammars and $\mathcal{L}(G)$ is the family of all languages.

(*c*) Let $G=(\{\sigma, a\}, \{a\}, P, \sigma)$, with $P=\{\sigma\to\sigma\sigma, \sigma\to a\}$. Then $\mathcal{G}(G)$ is the family of all grammar in Chomsky binary normal type and $\mathcal{L}(G)$ is the family of all context-free languages.

Results involving just $\mathcal{G}(G)$ or relations between grammars, such as "is an interpretation of", may be viewed as grammar theory. Results concerned with grammatical families may be either grammar theory or language theory, depending on the emphasis.

Finally we have:

Definition. Grammar forms G_1 and G_2 are said to be *strongly equivalent* if $\mathcal{G}(G_1)=\mathcal{G}(G_2)$, and *(weakly) equivalent* if $\mathcal{L}(G_1)=\mathcal{L}(G_2)$.

Thus strong equivalence is a grammar concept, while equivalence may be either a grammar or language concept.

The notion of interpretation given above is the most general that has been seriously considered. On the other hand, there are numerous restrictions on interpretations, leading to such kinds as nondecreasing,[4] length preserving,[5] strict,[6] etc. For each such kind of interpretation x, one may speak of *strong x-equivalence* and *(weak) x-equivalence*, meaning that $\mathcal{G}_x(G_1)=\mathcal{G}_x(G_2)$ and $\mathcal{L}_x(G_1)=\mathcal{L}_x(G_2)$, respectively, $\mathcal{G}_x(G_1)$ being the family of grammars obtained from x-interpretations of G_1 and $\mathcal{L}_x(G_1)$ being the family of languages $\{L(G)/G \text{ in } \mathcal{G}_x(G_1)\}$.

In presenting our survey of grammar form theory, we shall divide the results into five categories. These are grammar, language, decidability, complexity, and applications. As will be noted, some of the results fit into more than one category. In view of the nonmathematical nature of the applications and the mathematical nature of this audience, I shall not report on applications.

§ 2. Grammar theory

The results here are essentially of two kinds. The first involves the notion of "is an interpretation of", while the second concerns normalization theorems, i.e., results such as: For each grammar form with properties $A, B, \ldots$ there exists an equivalent grammar form with properties $P, Q, \ldots$

In [3] it was shown that the relation "is an interpretation of" is transitive. In [10] it was proved that modulo strong equivalence, all grammar forms under "is an

[4] For each element a in Σ, $\mu(a)$ is ε-free.

[5] For each element a in Σ, $\mu(a)$ is a finite subset of Σ_∞.

[6] μ is length preserving, and $\mu(a)\cap\mu(b)=\emptyset$ for all $a\neq b$ in Σ.

interpretation of" form a distributive lattice. Indeed, the existence of a glb for two grammar forms has an interesting restatement as: For all grammar forms G_1 and G_2, there exists a grammar form G_3 such that $\mathcal{G}(G_1)\cap\mathcal{G}(G_2)=\mathcal{G}(G_3)$. In [11], a new operator Q on a grammar form G is defined, yielding a family of grammars. Specifically, $Q(G)=\{G_I/I$ a quasi-interpretation of $G\}$, where a *quasi-interpretation* of a grammar form $G=(V, \Sigma, P, \sigma)$ is a 5-tuple $I=(\mu, V_I, \Sigma_I, P_I, S_I)$ satisfying

(i) μ is a substitution on V^* such that $\mu(a)$ is a finite subset of Σ_∞^* for each element a in Σ and $\mu(\xi)$ is a finite subset of $V_\infty-\Sigma_\infty$ for each ξ in $V-\Sigma$;

(ii) $P_I=\mu(P)$;

(iii) S_I is in $\mu(\sigma)$; and

(iv) $G_I=(V_I, \Sigma_I, P_I, S_I)$ is a grammar for which $V_I(\Sigma_I)$ contains each symbol (terminal) occurring in P_I.

Two results [11] involving $Q(G)$ are: For each grammar form G, $\mathcal{G}Q(G)=$ $=Q\mathcal{G}(G)$, and the collection of all families $\mathcal{G}(G')$, G' in $Q(G)$, is finite.

An outstanding open question is the following: Let G be a grammar form and $\mathcal{L}\subseteqq\mathcal{L}(G)$ a grammatical family. Is $\mathcal{L}$ in the class $\{\mathcal{L}(G_I)/I$ an interpretation of $G\}$? In other words, do all interpretation grammars of G, when viewed as grammar forms, yield all grammatical subfamilies of $\mathcal{L}(G)$? Analogous questions hold if interpretation is replaced by x-interpretation, x some "reasonable" kind of interpretation.

An open topic suggested by the Q operator is the following: Find different operators $\mathcal{U}$ on grammar forms G so that

(i) $\mathcal{U}(G)$ is a family of grammars, and

(ii) $\mathcal{U}$ has nice properties vis-a-vis operators already specified, e.g., with $\mathcal{G}$ and Q.

Onc would hope that there are a whole host of different operators yielding a variety of new relations and insights. Of special interest would be operators suggested by well-known transformations of grammars in, say compiler theory.

Turning to normalization results we have the following, proved in [3]: Each grammar form has an equivalent, completely reduced[7] sequential grammar form.

Indeed, one might think of a large class of normalization problems thusly: Let P be a property about grammars, e.g., unambiguity. Find grammar forms G with the property: There exists a grammar form G' so that $\mathcal{L}(G)=\{L(G_I)/G_I$ in $\mathcal{G}(G')$, G_I has property $P\}$.

There are many variations to the above stated canonical type problem. Consider this result [7]. If G is an unambiguous grammar form, then $\mathcal{L}_{\text{strict}}(G)=$ $=\{L(G_I)/G_I$ in $\mathcal{G}_{\text{strict}}(G)$, G_I unambiguous$\}$. Thus, there are "sufficiently many" unambiguous strict interpretations of an unambiguous grammar form to yield all strict interpretation languages.

Finally, in [14] various kinds, x, of interpretations of a form are studied from the viewpoint of conflict freeness (as used in compiling). For example, let $G=(V, \Sigma, P, \sigma)$ be a grammar form with the property that for each variable ξ there is a non ε terminal word w such that $\xi\overset{+}{\Rightarrow}w$. Then the following three conditions occur simultaneously:

(1) $\mathcal{G}(G)$ is conflict free (i.e., each grammar in $\mathcal{G}(G)$ is conflict free).

[7] A grammar form $G=(V, \Sigma, P, \sigma)$ is *completely reduced* if (i) G is reduced, (ii) there are no variables α and β such that $\alpha\to\beta$ is in P, and (iii) for each variable α in $V-(\Sigma\cup\{\sigma\})$ there exist x and y in Σ^*, $xy\neq\varepsilon$, such that $\alpha\to x\alpha y$ is in P.

(2) $\mathscr{G}_{\text{nondecreasing}}(G)$ is conflict free.

(3) G is separated (that is, for each rule $\xi \to w$ in P, w is in $(V-\Sigma)^* \cup \Sigma^*$) and whenever a rule $\xi \to \gamma$ is in P, with γ in $(V-\Sigma)^+$, then γ is in $V-\Sigma$.

Given a grammar form G, $\mathscr{G}_{\text{strict}}(G)$ is conflict free if and only if G is conflict free. Characterization results are presented on a grammar form in order for it to have a strongly ($x-$) equivalent conflict free grammar form, where x is strict, length preserving, and nondecreasing, respectively. It is also shown that every grammar form has an equivalent conflict free grammar form.

§ 3. Language theory

We now review some language theory results. Since language theory itself is so vast, this section could easily dominate all the others. In addition, it is very easy, considering our experience, to phrase innumerable questions about grammar forms which have a language theory flavor. While one cannot stop "progress", I personally believe it is not in the best interests of grammar form theory to exploit grammar forms for the purpose of language interests. The real aim of grammar form theory should be to develop new ideas, insights, questions, etc. about *grammar* concepts.

In § 1, examples were given to show that the regular sets, the linear languages, and the context-free languages are grammatical families. In [3], characterizations on G were given in order that $\mathscr{L}(G)$ be

(1) $\mathscr{R}$, the family of regular sets,

(2) $\mathscr{L}_{\text{lin}}$, the family of linear languages, and

(3) $\mathscr{L}_{CF}$ the family of context free languages.

For (3), the if and only if is quite interesting, namely that G be an expansive grammar in the classical language theory sense. From this it follows that each grammatical family $\mathscr{L}(G) \neq \mathscr{L}_{CF}$ contains only derivation bounded languages. Thus, the one-counter languages are not a grammatical family. This might explain why no "simple" type of context-free-like grammar is around to describe these languages.

Whenever one has a family of languages, it makes sense to investigate its closure properties. For grammar forms we have the surprising result [3] that if G is nontrivial, i.e., $L(G)$ is infinite, then $\mathscr{L}(G)$ is a full principal semi-AFL. The converse, of course, is not true. As mentioned above, the one-counter languages are not a grammatical family. Neither is the full principal semi-AFL generated by $\{a^n b^n / n \geqq 1\}$. In connection with the above semi-AFL result there is a cluster of open questions concerning grammars G such that $L(G)$ is a full generator for $\mathscr{L}(G)$. For example, what are some necessary and sufficient conditions on G, or what are just some useful sufficient conditions? The reader is cautioned to be careful. There are many pitfalls. My favorite is this: $G = (\{\sigma, a, b\}, \{a, b\}, \{\sigma \to a\sigma b, \sigma \to ab\}, \sigma)$ is a form for which $\mathscr{L}(G) = \mathscr{L}_{\text{lin}}$. On the other hand, $L(G) = \{a^n b^n / n \geqq 1\}$, which is not a full generator for $\mathscr{L}_{\text{lin}}$.

One of the major operations in language theory is that of substitution. It is thus natural to try to define the substitution of one grammar form into another. This can be done as follows: For grammar forms G and G', let Sûb (G, G') be the form obtained by substituting the start variable of G' for every occurrence of a terminal in the productions of G. This yields [13] the obvious result desired, namely, if

G is nontrivial then for every grammar form[8] G', $\mathscr{L}(\text{Sûb}(G,G'))=\text{Sûb}(\mathscr{L}(G),\mathscr{L}(G'))$. Now it is known that if $\mathscr{L}$ is a full semi-AFL, then Sûb $(\mathscr{R},\mathscr{L})$ is a full AFL. Since the grammar form with rules $\sigma\to a\sigma$, $\sigma\to a$ yields $\mathscr{R}$, it follows that for each grammar form $G'=(V',\Sigma',P',\sigma')$, the form Sûb $(G,G')=(V'',\Sigma',P'',\sigma')$ where $P''=P'\cup\cup\{\sigma\to\sigma'\sigma,\sigma\to\sigma'\}$ yields the full AFL generated by $\mathscr{L}(G')$.

Earlier, we noted that for each nontrivial grammar form G, $\mathscr{L}(G)$ is a full principal semi-AFL. It remains an open problem to characterize "internally" those full semi-AFL which are grammatical families. However, we can given "external" characterizations of such semi-AFL. These characterizations are similar in spirit to the Kleene theorem for regular sets, in that they describe the collection of almost all grammatical families in terms of a few elementary ones and composition under some basic operations. We elaborate. For sets $\mathscr{L}_1$ and $\mathscr{L}_2$ of languages, let

$$\mathscr{L}_1\vee\mathscr{L}_2=\{L_1\cup L_2/L_1 \text{ in } \mathscr{L}_1, L_2 \text{ in } \mathscr{L}_2\}$$

and

$$\mathscr{L}_1\odot\mathscr{L}_2=\left\{\bigcup_{i=1}^{k} L_{1i}L_{2i}/k\geqq 1,\text{ each } L_{1i} \text{ in } \mathscr{L}_1,\text{ each } L_{2i} \text{ in } \mathscr{L}_2\right\}.$$

Let $\hat{\mathscr{F}}$ be the full AFL operator, i.e., for each family $\mathscr{L}$ of languages let $\hat{\mathscr{F}}(\mathscr{L})$ be the smallest full AFL containing $\mathscr{L}$. Finally, for all sets $\mathscr{L}_a,\mathscr{L}_b,\mathscr{L}_c$ of languages, let $\mathscr{T}(\mathscr{L}_a,\mathscr{L}_b,\mathscr{L}_c)=\{\tau(L)/L=L(G),\ G=(V_1,A\cup B\cup C,P,\sigma)$ is a split linear grammar,[9] τ is a substitution on $(A\cup B\cup C)^*$ such that $\tau(x)$ is in $\mathscr{L}_a$ if x is in A, $\tau(x)$ is in $\mathscr{L}_b$ if x is in B, and $\tau(x)$ is in $\mathscr{L}_c$ if x is in $C\}$. There are two characterization results about the grammatical families [4]. The first is: The collection of all grammatical families not $\{\emptyset\}$ and not $\mathscr{L}_{CF}$ is the smallest collection of sets of languages containing $\mathscr{L}_\varepsilon=\{\{\varepsilon\}\}$ and $\mathscr{L}_{\text{fin}}=\{$all finite languages$\}$ and closed under $\vee$, $\odot$, and $\mathscr{T}$. The second is: The collection of all nontrivial grammatical families not $\mathscr{L}_{CF}$ is the smallest collection of sets of languages containing $\mathscr{R}$ and closed under $\vee$, $\odot$, $\mathscr{T}$, and $\hat{\mathscr{F}}$.

At the beginning of this section it was mentioned that each grammatical family not $\mathscr{L}_{CF}$ is a family of derivation bounded languages. As any language theorist knows, there is a close analogy between derivation bounded languages and nonterminal bounded languages. Question — are the nonterminal bounded languages lurking in the grammarform bushes? Answer — yes, if you look for them. Let us call a grammar form $G=(V,\Sigma,P,\sigma)$ *sequentially ultralinear* if

(i) it is sequential, and

(ii) whenever $\xi\to\alpha\xi\beta$ is in P, α and β in V^*, then $\alpha\beta$ is in Σ^*.

Call a grammatical family *ultralinear* if it is generated by some sequentially ultralinear grammar form. The following result has been established [6]. The three statements:

[8] For two families $\mathscr{L}_1$ and $\mathscr{L}_2$ of languages, Sûb $(\mathscr{L}_1,\mathscr{L}_2)=\{\tau(L_1)/L_1$ in $\mathscr{L}_1$, τ is a substitution on L_1 such that $\tau(a)$ is in $\mathscr{L}_2$ for every symbol $a\}$.

[9] A *split linear grammar* is a linear grammar $G=(V_1,\Sigma_1,P_1,\sigma_1)$ such that there exist disjoint sets A, B, C with the following properties: (1) $\Sigma_1=A\cup B\cup C$. (2) Every terminal production is of the form $\xi\to c$ for some ξ in $V_1-\Sigma_1$ and c in C. (3) Every production which is not a terminal one is of the form $\xi\to a\xi'$ for some ξ,ξ' in $V_1-\Sigma_1$ and a in A or $\xi\to\xi'b$ for some ξ,ξ' in $V_1-\Sigma_1$ and b in B.

(a) $\mathscr{L}$ is a nontrivial ultralinear grammatical family;
(b) $\mathscr{L}$ is a nontrivial grammatical family of nonterminal bounded languages; and
(c) $\mathscr{L}$ can be built up from $\mathscr{R}$ by a finite sequence of applications of $\odot$, $\vee$, and $[\,]$, where $[\mathscr{L}]=\mathscr{T}(\mathscr{R}, \mathscr{L}, \mathscr{R})$;

are equivalent. Thus, a relatively simple class of grammar forms gives rise to a rather natural class of families of languages.

A rather popular topic in language theory is that of control sets. In [16, 17] Greibach has presented a number of results in which control sets play a leading role. The following is a sample. Let G be a nontrivial left derivation bounded grammar form with left derivation bound k. Then there is a nontrivial equivalent grammar form $G_0=(V_0, \Sigma_0, P_0, \sigma_0)$, left derivation bounded with left derivation bound k, such that for each finite alphabet Σ, $\{L\cap\Sigma^*/L \text{ in } \mathscr{L}(G)\}$ consists of all languages obtained by using regular sets as control sets for leftmost derivations over $\tau_\Sigma(G_0)$. $[\tau_\Sigma(G_0)=(V_0, \Sigma, \tau_\Sigma(P_0), \sigma_0)$, where τ_Σ is the substitution on V_0^* defined by $\tau_\Sigma(\xi)=\{\xi\}$ for each ξ in $V_0-\Sigma_0$ and $\tau_\Sigma(a)=\Sigma\cup\{\varepsilon\}$ for all a in Σ_0.$]$

§ 4 Decidability

There are a number of different decidability results. We shall mention a fair sampling.

It is solvable [3] to determine whether or not, given an arbitrary grammar G' and grammar form G, there is an interpretation I of G such that $G'=G_I$. Also, the strong equivalence problem is solvable. One question that has been open since the beginning of grammar form theory is the decidability of (weak) equivalence. That is, can one tell for arbitrary grammar forms G_1 and G_2 whether $\mathscr{L}(G_1)=\mathscr{L}(G_2)$? Even though the problem is standard in situations of this kind, nevertheless, its solution here seems to be of importance since it seems to be related to several questions involving two or more grammatical families. For example, is $\mathscr{L}(G_1)\cap\mathscr{L}(G_2)$ always a grammatical family? Given a context-free language L, does there exist a smallest grammatical family containing L?

Research is currently underway with respect to the decidability of equivalence. The author, in conjunction with JONATHAN GOLDSTINE and EDWIN H. SPANIER, has reduced the problem to about ten inclusion problems involving the operators $\vee$, $\odot$, $\mathscr{F}$, and $\mathscr{T}$. We think we have resolved all the cases (thereby settling the decidability in the affirmative). However, until *all* the details have been written, we are making no claim. We hope to be able to announce the answer within three months (say December 1, 1977).

A special case of the equivalence problem has been resolved affirmatively. In [6] it is shown that for any two sequentially ultralinear grammar forms G_1 and G_2, it is solvable to determine if $\mathscr{L}(G_1)\subset\mathscr{L}(G_2)$, and therefore if $\mathscr{L}(G_1)=\mathscr{L}(G_2)$. The proof is quite involved, and consists of showing that the operations of $\odot$, $[\,]$, and $\vee$ applied to $\mathscr{R}$, when suitably restricted in combination, are intimately determined by the end ultralinear grammatical family. Indeed, and this is a surprising fact, there is an essentially unique canonical representation of each nontrivial ultralinear grammatical family in terms of "semibracketed expressions", namely, certain combinations of $\mathscr{R}$, $\odot$, $\vee$, and $[\,]$.

In [7], certain decidability results are established for strict interpretations of unambiguous grammar forms. Specifically, for each unambiguous grammar form and each positive integer k, it is decidable whether

(a) an arbitrary strict interpretation grammar is k-ambiguous;

(b) for any k languages $L_1, \ldots, L_k$ generated by[10] compatible strict interpretation grammars, (i) $\bigcap_{i=1}^{k} L_i$ is empty, (ii) $\bigcap_{i=1}^{k} L_i$ is finite, (iii) $\bigcup_{i=1}^{k} L_i$ is infinite; and

(c) for any two languages L_1 and L_2 generated by compatible strict interpretation grammars, (i) $L_1 \subseteqq L_2$ and (ii) $L_1 = L_2$.

§ 5. Complexity

While some work has been done on complexity, this essentially is an area which has received only modest attention. Indeed, the summary given below is basically the same as given in section 5 of [5], with the inclusion of some material from [7].

In [10], it is shown that for each grammar form G there exists an "essentially unique" strongly equivalent form G' with the fewest number of productions possible. Furthermore, G' can always be found with its productions a subset of those of G.

Complexity of derivations is studied in [9]. For each grammar form G and each grammar G' in $\mathcal{G}(G)$, the complexity function $\Phi_{G'}$ is defined for each word x in $L(G')$ as the number of steps in a minimal G'-derivation of x. It is proved that derivations may also be speeded up by any constant factor n, in the sense that for each positive integer n, an equivalent grammar G'' in $\mathcal{G}(G)$ can be found so that $\Phi_{G''}(x) \leqq \frac{|x|}{n}$ for all large words x.

In [10] grammar forms are compared for their efficiency in representing languages, as measured by the sizes (i.e., total number of symbols, number of variable occurrences, number of productions, and number of distinct variables) of interpretation grammars. Right- and left-linear forms are essentially equal in efficiency for every regular set. Each form for the regular sets provides at most polynomial improvement over right-linear form. Moreover, any polynomial improvement is attained by some such form, at least on certain languages. Greater improvement for some languages is possible with forms expressing larger classes of languages than the regular sets. However, there are some languages for which no improvement over right-linear form is possible. A similar set of results holds for forms expressing exactly the linear languages. On the other hand, only linear improvement can occur for forms expressing $\mathcal{L}_{CF}$.

There is one more place where complexity has been considered. This is in

[10] Strict interpretations $I_j = (\mu_{I_j}, V_{I_j}, \Sigma_{I_j}, S_{I_j})$, $j = 1, \ldots, k$, $k \geqq 2$, of a grammar form (V, Σ, P, σ) are called *compatible* if $\left(\bigcup_{j=1}^{k} \mu_{I_j}(x)\right) \cap \left(\bigcup_{i=1}^{k} \mu_{I_i}(y)\right) = \emptyset$ for all x, y in V wih $x \neq y$.

parsing. While parsing can be regarded as an application, for the present purpose I shall catalogue it under complexity. The first result is from [1]. Let G be an arbitrary unambiguous grammar form. Suppose there is a function $t(n)$, $n \geqq 0$, and a parsing procedure M_G for G which, for each word ω, in $t(|\omega|)$ steps, parses ω if in $L(G)$ and rejects ω if not in $L(G)$. Then for each strict interpretation $I=(\mu_I, G_I)$ of G, there exist a parsing procedure M_I for $G_I=(V_I, \Sigma_I, P_I, S_I)$ and a constant c with the following property: For each word w in Σ_I^*, M_I, in $c \cdot t(|w|)$ steps, accepts w if w is in $L(G_I)$ and rejects w if it is not in $L(G_I)$. This result has been generalized in [7]. Specifically, let $G=(V, \Sigma, P, \sigma)$ be an arbitrary grammar form and suppose there is a parsing method M_G for G and a function $t(n)$, $n \geqq 0$, such that for each word of length n, M_G outputs all leftmost derivations of that word in at most $t(n)$ steps. Let $I=(\mu, V_I, \Sigma_I, P_I, S_I)$ be a strict interpretation of G. Then there exists a parsing procedure M_I for G_I and a constant c such that for each word w in Σ_I^*, in $c \cdot t(|w|)$ steps, M_I accepts w if in $L(G_I)$ and rejects w if not in $L(G_I)$. Furthermore, if $p(n)$, $n \geqq 0$, is such that for each word of length n in $L(G_I)$ there are no more than $p(n)$ equally-shaped derivations[11] of that word, then M_I yields, in $c \cdot t(|w|)$ steps, all leftmost G_I-derivations of w.

§ 6. Grammar forms which are not context-free

In the present section, I shall discuss grammar forms which are not necessarily context-free. [The definitions of interpretation, $\mathscr{L}(G)$, etc. carry through in the obvious way.]

The original definition of grammar form, as given in [3], was for arbitrary phrase structure grammars. Due to the scarcity of results in such a general situation, the investigation was quickly limited to context-free grammars and has stayed that way since. At present, with the exception of the first part of [3], the only results on arbitrary grammar forms are in [18]. The basic, original question, and it is still unresolved, is this: Are there any grammar forms G such that

(*) $\mathscr{L}(G) \subseteqq \mathscr{L}_{CF}$ is false and $\mathscr{L}(G) \neq \mathscr{L}_{RE}$, $\mathscr{L}_{RE}$ being the family of recursively enumerable sets?

In 1972, I mentioned this problem to my associate Dr. Gene F. Rose. He struggled with (*), on and off, for several years, to no avail. [That means that the question is difficult.] His opinion was that the answer to (*) was probably no. This opinion is also shared by the authors of [18], as is noted in their abstract. Some progress was made in [18], since it was shown there that the answer to (*) is no when the grammar form has exactly one nonterminal.

Even if the answer to (*) turns out negative, the subject of non context-free grammar forms should be a fertile field of study. All interpretations need not be studied. One could examine appropriate subclasses. [An analogous situation arises with the family of context-sensitive languages. It is not discarded just because its closure under arbitrary homomorphism is $\mathscr{L}_{RE}$.] In fact, a start on this aspect has

[11] Two derivations are *equally shaped* if their parse trees are equally shaped. Two derivation trees are *equally shaped* if each tree can be obtained from the other by relabeling nonmaximal nodes.

already been done in [18]. A number of different, restricted types of interpretations of non context-free forms are considered, and then used to characterize several well-known language families between $\mathscr{L}_{CF}$ and $\mathscr{L}_{RE}$, such as EOL, ETOL, matrix, and scattered languages. Much remains to be done.

§ 7. Future development

The discussion up to now has been on grammar forms. I would like to speak about the general notion of a form as a method of studying when one graphlike structure looks like another.

As we all know, there is a considerable body of knowledge, under the title "*L* systems," of context-free grammars in which parallel derivation occurs, that is, at each step each symbol in the string is replaced. During the past two years the concept of an *L*-form (forms applied to *L*-systems) has been studied [L1—L10]. The results themselves are of no concern to the present discussion. What is of interest is that the notion of form has been carried over to this graphlike structure, with fruitful consequences arising.

Recently, a study was made of pushdown acceptor forms (pda forms) [14]. The aim here is to get a right definition of when one pda looks like another. If one thinks of an input symbol to a pda as a terminal and a state of a pda as a nonterminal, then input symbols are replaced by finite sets of input strings and states by finite sets of states. In addition, distinct states go into disjoint sets of states. But how should one handle replacement of symbols on the auxiliary storage? The key is to regard auxiliary symbols as additional storage. Since states (which are storage) are replaced by finite sets of states (with the disjointness property), pushdown symbols should be replaced by finite sets of pushdown symbols (with the disjointness property). The main question considered for pda forms is what are the resulting families of languages? Because context-free languages coincide with pda languages, the obvious answer would appear to be the class of all grammatical families. And indeed, this is what does happen! However, the proof is quite involved. In any case, the coincidence of the two classes of families is an indication of the "correctness" of the abstraction mode.

Currently, in conjunction with Dr. E. F. Schmeichel, I am working on "graph forms" and "looks like" for graphs. The idea is simple. Nodes and edges in a graph are like nonterminals. One must be careful to see that linkage corresponds. Specifically, we have:

Definition. Let $G=(N, E)$ be a (finite) graph. An *interpretation* of G is a triple $I=(\mu, N_I, E_I)$, where μ is a function on $N\cup E$ such that

(i) $\mu(v)$ is a finite set of nodes for each v in N, with $\mu(v_1)\cap\mu(v_2)=\emptyset$ for $v_1\neq v_2$,

(ii) $N_I\subseteqq \bigcup_{v \text{ in } N} \mu(v)$, and

(iii) $E_I\subseteqq \bigcup_{e \text{ in } E} \mu(e)$, with $\mu(v_1, v_2)=\mu(v_1)\times\mu(v_2)$ for each edge $e=(v_1, v_2)$.

For each graph form G let $\mathscr{G}(G)=\{G_I/I \text{ an interpretation of } G\}$.

The investigation here is in its infancy and results obtained to date are scattered. In view of the similarity between interpretations for grammar forms, *L*-forms, pda forms, and graph forms, it seems highly likely that other graphlike structures can be treated from the form perspective. Situations that readily come to mind are: Petri nets, pattern theory, data bases,[12] data types,[13] security models, various types of acceptors. The key in each instance is to determine what "looks like" (i.e., the μ function) is to mean for those features of graphlike structures which are not analogous to variables in a grammar. There does not seem to be any straightforward way of doing this. Rather, insight and trial-and-error appear to be the main techniques. The benefits to be accrued from a successful model for almost any kind of graphlike structure are a strong incentive.

Abstract

The present paper gives an overview of grammar form theory 1977. Concepts, results, and open questions are considered. In addition, general philosophy and future directions are expounded.

UNIVERSITY OF SOUTHERN CALIFORNIA
LOS ANGELES, CALIFORNIA, USA

Bibliography on grammar forms

[1] BERTSCH, E., An observation on relative parsing time, *JACM*, v. 22, 1975, pp. 493—498.
[2] BLATTNER, M. and S. GINSBURG, Position restricted grammar forms, in preparation.
[3] CREMERS, A. and S. GINSBURG, Context-free grammar forms, *JCSS*, v. 11, 1975, pp. 86—116.
[4] CREMERS, A. S. GINSBURG and E. H. SPANIER, The structure of context-free grammatical families, *JCSS*, v. 15, 1977, pp. 262—279.
[5] GINSBURG, S., A survey of context-free grammar forms, in *Formal Languages and Programming*, (R. Aguilar, ed.) North-Holland, 1976.
[6] GINSBURG, S. and J. GOLDSTINE, Ultralinear grammatical families, in preparation.
[7] GINSBURG, S., B. LEONG, O. MAYER and D. WOTSCHKE, On strict interpretations of grammar forms, in preparation.
[8] GINSBURG, S. and N. LYNCH, Size complexity in context-free grammar forms, *JACM*, v. 23, 1976, pp. 582—598.
[9] GINSBURG, S. and N. LYNCH, Derivation complexity in context-free grammar forms, *SIAM J. on Computing*, v. 6, 1976, pp. 123—138.
[10] GINSBURG, S. and H. MAURER, On strongly equivalent context-free grammar forms, *Computing*, v. 16, 1976, pp. 281—291.
[11] GINSBURG, S. and H. MAURER, On quasi-interpretations of grammar forms, *Computing* v. 19, 1977, pp. 141—147.
[12] GINSBURG, S. and E. ROUNDS, Dynamci syntax specification using grammar forms, *IEEE Trans. on Software Engineering*, v. SE—4, 1978, pp. 44—55.
[13] GINSBURG, S. and E. H. SPANIER, Substitution of grammar forms, *Acta Mathematica*, v. 5, 1975, pp. 377—386.
[14] GINSBURG, S. and E. H. SPANIER, Pushdown acceptor forms, to appear in *Theoretical Computer Science*.
[15] GINSBURG, S. and D. WOOD, Simple precedence relations in grammar forms, submitted for publication.
[16] GREIBACH, S., Control sets on context-free grammar forms, *Journal of Computer and System Sciences*, v. 15, 1977, pp. 35—98.

[12] One of my doctoral students is now investigating this.

[13] I have been looking at this, in conjunction with DR. JOHN GUTTAG. There is nothing to report on as yet.

[17] Greibach, S., Comments on universal and left universal grammars, context-sensitive languages and context-free grammar forms, submitted for publication.

[18] Maurer, H., M. Penttomen, A. Salomaa and D. Wood, On non context-free grammar forms, submitted for publication.

[19] Maurer, H. and D. Wood, On grammar forms with terminal context, *Acta Informatica*, v. 6, 1976, pp. 397—401.

[20] Walter, H., Grammatik und Sprachfamilien I, Report F. G. Automatentheorie und Formale Sprachen, T. H. Darmstadt, 1975.

[21] Walter, H., Grammatik und Sprachfamilien II, Report F. G. Automatentheorie und Formale Sprachen, T. H. Darmstadt, 1975.

[22] Walter, H., Grammatik und Sprachfamilien III, Report F. G. Automatentheorie und Formale Sprachen, T. H. Darmstadt, 1976.

[23] Walter, H., Grammatik und Sprachfamilien IV, Report F. G. Automatentheorie und Formale Sprachen, T. H. Darmstadt, 1976.

[24] Walter, H., Grammarforms und Grammarhomomorphisms, *Acta Informatica*, v. 7, 1976, pp. 75—93.

[25] Walter, H., Structural equivalence of context-free grammar forms, to appear.

Bibliography on *L*-forms

[L1] Culik, K. II and H. A. Maurer, Propagating chain-free normal forms for EOL systems, submitted for publication.

[L2] Culik, K. II, H. A. Maurer and T. Ottman, Two letter complete EOL forms, submitted for publication.

[L3] Culik, K. II, H. A. Maurer, T. Ottman, K. Kuohonen and A. Salomaa, Isomorphism, form equivalence and sequence equivalence of pdol forms, submitted for publication.

[L4] Hule, H., H. A. Maurer and T. Ottman, OL forms, submitted for publication.

[L5] Maurer, H. A., A. Salomaa and D. Wood, EOL forms, *Acta Mathematica*, v. 8, 1977, pp. 75—96.

[L6] Maurer, H. A., A. Salomaa and D. Wood, ETOL forms, submitted for publication.

[L7] Maurer, H. A., A. Salomaa and D. Wood, Good EOL forms, submitted for publication.

[L8] Maurer, H. A., A. Salomaa and D. Wood, Uniform interpretations of *L*-forms, submitted for publication.

[L9] Maurer, H. A., T. Ottman and A. Salomaa, On the form equivalence of *L*-forms. *Theoretical Computer Science*, v. 4, 1977, pp. 199—225.

[L10] Skyum, S., On good ETOL forms, submitted for publication.

(Received Sept. 5, 1977)