THE INFLUENCE OF PRODUCTIONS ON DERIVATIONS AND PARSING

(Extended Abstract)

Benton L. Leong and Detlef Wotschke
Computer Science Department
The Pennsylvania State University
University Park, PA. 16802

ABSTRACT

The concept of grammar forms [4,5] provides evidence that there seems to be no way to base the definitions of many grammar types used in parsing and compiling solely on the concept of productions.

Strict interpretations, as introduced in [3,5], of unambiguous or LR(k) grammar forms generate unambiguous or LR(k) languages, respectively. This is not true in the LL(k) case.

It is decidable whether a strict interpretation of an unambiguous grammar form is unambiguous. For any two compatible strict interpretations G_1 and G_2 of an unambiguous grammar form it is decidable whether $L(G_1) \subseteq L(G_2)$, $L(G_1) \cap L(G_2) = \emptyset$, finite, or infinite.

For every grammar form F_1 there exists a grammar form F_2 such that the grammatical family of F_1 under unrestricted interpretations is equal to the grammatical family of F_2 under strict interpretations.

I. INTRODUCTION

The definitions of most grammar types that are used for parsing and compiling depend on the concept of a derivation. For example, the definition of an LR(k) grammar depends on rightmost derivations, while the definition of an LL(k) grammar depends on leftmost derivations. On the other hand, the definition of a linear context-free grammar, for instance, depends solely on the concept of a production. We will show in section II that the concept of grammar forms ([4], [5]) provides evidence that there seems to be no way to base the definitions of many grammar types used in parsing and compiling solely on the concept of productions instead of derivations.

Bertsch has shown in [3] that so-called strict interpretations [3,5] of unambiguous grammar forms require essentially the same parsing time, up to a multiplicative constant c, as the underlying unambiguous grammar form. One of the conditions, which is quite restrictive and essential for the proof, is missing in the definition of a strict interpretation in [3]. So the question arises: how restricted are strict interpretations. We will prove in section III that strict interpretations of unambiguous or LR(k) grammar forms generate unambiguous or LR(k) languages, respectively. This is not so in the LL case. It is decidable whether a strict interpretation of an unambiguous grammar form is unambiguous. For any two "compatible" strict interpretations G_1 and G_2 of an unambiguous grammar form it is decidable whether $L(G_1) \subseteq L(G_2)$ and whether $L(G_1) \cap L(G_2)$ is empty, finite, or infinite.

In section IV we will compare grammar forms with

strict and unrestricted interpretations. Specifically we will show that for every grammar form F_1 there exists a grammar form F_2 such that the grammatical family of F_1 under unrestricted interpretations is equal to the grammatical family of F_2 under strict interpretations. There are cases where F_2 has to be ambiguous although F_1 is unambiguous.

A few open problems will be listed in section V.

II. THE INFLUENCE OF PRODUCTIONS ON DERIVATIONS

For many years a considerable amount of time and effort has been spent defining and investigating grammars which are useful as models for programming languages. Among these grammars are, to name only a few, LR(k) grammars, LL(k) grammars, bounded-context grammars, (m,n) precedence grammars, simple precedence grammars (cf. [1] as a survey reference). Although all these models differ quite significantly from each other in their definitions as well as in their parsing techniques, almost all of them have one thing in common: their definitions rely heavily on the concept of a derivation. We omit quoting the definitions of LR(k)-, LL(k)-, simple precedence grammars, etc. But clearly part of the definition of an LR(k) grammar is the underlying rightmost derivation. Similarly the definition of an LL(k) grammar depends on the underlying leftmost derivation. As a consequence, testing a grammar for LR(k)-ness, LL(k)-ness (for a given k) involves more, at least in the general case, than a mere checking of whether each production in the given production table is of a certain form. In many instances, testing a grammar for LR(k)-ness or LL(k)-ness (for a given k) requires quite an involved computation.

There are, of course, other ways to define grammars in general, and specific types of context-free grammars in particular. For example, it is extremely easy to define and understand the notion of a linear context-free grammar. Likewise, it is a trivial task to test whether a given grammar is a linear context-free grammar, a regular grammar, a context-free grammar, etc.

The immediate question comes up whether it is really necessary to define LR(k) grammars, LL(k) grammars, etc. in terms of derivations rather than just productions. In other words, do there or do there not exist methods to define LR(k)-, LL(k)-, simple precedence grammars, etc. by merely placing certain restrictions on the production type allowed. This question is extremely important since restrictions on productions are generally easier to understand than restrictions on derivations. Many proofs (e.g., the equivalence of deterministic pushdown automata and LR(1) grammars) would hopefully become simpler if the grammar definitions were based on production types rather than on derivation types. It was often assumed that such a "production-characterization" is probably not possible, partly because of the following reason: it is undecidable, for example, whether a given context-free grammar is an LL(k) grammar or LR(k) grammar for any k ([1]). It is therefore impossible to find a characterization (in terms of production-types) of LR(k)-, LL(k) grammars etc. Otherwise, one could decide whether or not a given grammar is LL(k) or LR(k) for any k by simply checking each production for "characteristic" properties, assuming, of course, that testing for these "characteristic" properties is a decidable task.

However, it is decidable whether a grammar is LR(k), LL(k) for any fixed k, and it is also decidable whether a grammar is a simple precedence grammar [1]. So it is not at all clear whether or not there exist characterizations of LL(k)-, LR(k) grammars (for fixed k), or simple precedence grammars in terms of productions. Before one can start investigations of this kind one has to have a well-defined concept of what a characterization in terms of productions should be. Such a concept, that of a grammar form, has been established by Cremers and Ginsburg ([4],[5]).

Definition II.1: A (context-free) grammar form is a 6-tuple $F = (\mathcal{V},\mathcal{S},V,\Sigma,P,S)$ where

1) $\mathcal{V}$ is an infinite set of abstract symbols,
2) $\mathcal{S}$ is an infinite subset of $\mathcal{V}$ such that $\mathcal{V}-\mathcal{S}$ is infinite, and
3) $G_F = (V,\Sigma,P,S)$, called the form grammar (of F), is a (context-free) grammar ([6],[11]) with $V \subseteq \mathcal{V}$, $\Sigma \subseteq \mathcal{S}$, and $(V-\Sigma) \subseteq (\mathcal{V}-\mathcal{S})$.

We assume throughout this paper that $\mathcal{V}$ and $\mathcal{S}$ are fixed infinite sets satisfying (1) and (2) above. We will only consider grammar forms which have a context-free form grammar and will therefore omit the attribute "context-free" in many instances.

Definition II.2: An interpretation of a grammar form $F = (\mathcal{V},\mathcal{S},V,\Sigma,P,S)$ is a 5-tuple $I = (\mu,V_I,\Sigma_I,P_I,S_I)$, where

1) μ is a substitution on V^* such that
   i) $\mu(a)$ is a finite subset of $\mathcal{S}^*$ for each element a in Σ,
   ii) $\mu(A)$ is a finite subset of $\mathcal{V}-\mathcal{S}$ for each A in $V-\Sigma$, and
   iii) $\mu(A) \cap \mu(B) = \emptyset$ for each A and B in $V-\Sigma$ if $A \neq B$,
2) P_I is a subset of $\mu(P) = \bigcup_{\pi \in P} \mu(\pi)$, where $\mu(\alpha \to \beta) = \{u \to v \mid u \in \mu(\alpha),\ v \in \mu(\beta)\}$,
3) S_I is in $\mu(S)$, and
4) $\Sigma_I(V_I)$ contains all symbols in $\Sigma(V)$ which occur in P_I (together with S_I).

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

A grammar form gives rise to a family of grammars, which are structurally related to the form grammar, by means of interpretations of the form grammars.

For brevity, an interpretation I is frequently written as (μ,G_I). Since the productions in G_I are structurally related to the form grammar G_F, we sometimes will call G_F a "master grammar." Since $\mathcal{V}$ and $\mathcal{S}$ are fixed, we will use the phrase "an interpretation of G", where $G = (V,\Sigma,P,S)$, rather than "an interpretation of F", where $F = (\mathcal{V},\mathcal{S},V,\Sigma,P,S)$.

Definition II.3: For each grammar G, $\mathcal{G}(G) = \{G_I \mid I$ an interpretation of $G\}$ is called the family of grammars (of G) and $\mathcal{L}(G) = \{L(G_I) \mid G_I$ is in $\mathcal{G}(G)\}$ the grammatical family (of G). A collection $\mathcal{L}$ of languages is a grammatical family if $\mathcal{L} = \mathcal{L}(G)$ for some grammar G.

It was shown in [4] that the linear context-free languages, the regular languages, and the context-free languages all constitute grammatical families. For the sake of exposition we briefly list grammar forms for the above mentioned classes of languages.

1) Let G be a grammar with $P = \{S \to aS, S \to e\}$. Then clearly $\mathcal{L}(G)$ is the family of regular languages.
2) Let G be a grammar with $P = \{S \to aSa, S \to e\}$. Then $\mathcal{L}(G)$ is the family of all linear context-

free languages.

3) Let G be a grammar with $P = \{S \to SS, S \to a\}$, then $L(G)$ is the family of all context-free languages.

The above examples demonstrate that grammar forms are a reasonable model for production types.

Using this model of grammar forms it is very easy to show that many classes of grammars that possess interesting parsing properties do not form families of grammars and that it is therefore highly unlikely that their relevant features can be expressed in terms of productions. But first we restate the following theorem from [4]:

Theorem II.4: For each grammar G, $L(G)$ is closed under union, homomorphism, and intersection with regular sets.

We can now state:

Theorem II.5: Each of the following classes of grammars does not form a family of grammars:

1) LL(k) grammars for arbitrary k
2) LR(k) grammars for arbitrary k
3) LL(k) grammars for arbitrary, but fixed k
4) LR(k) grammars for arbitrary, but fixed k
5) Strong LL(k) grammars for any given k
6) Simple LR(k) grammars for any given k
7) Strict deterministic grammars
8) Simple precedence grammars
9) Uniquely invertible extended precedence grammars
10) Uniquely invertible weak precedence grammars
11) Simple mixed strategy precedence grammars
12) Mixed strategy precedence grammars
13) (1,1) bounded right context grammars
14) Bounded right context grammars.

Proof: Consider the following two grammars $G_1 = (V_1, \Sigma_1, P_1, S_1)$ and $G_2 = (V_2, \Sigma_2, P_2, S_2)$ where

$V_1 = V_2 = \{a,b,c,S\}$

$\Sigma_1 = \Sigma_2 = \{a,b,c\}$,

$P_1 = \{S \to aSb,\ S \to c\}$, and

$P_2 = \{S \to aSbb,\ S \to c\}$.

Clearly $L_1 = L(G_1) = \{a^n cb^n \mid n \geq 0\}$ and $L_2 = L(G_2) = \{a^n cb^{2n} \mid n \geq 0\}$. Let $\mathcal{G}_i$ denote the class of all grammars of type i) (i=1,...,14) as listed in the theorem. Thus $\mathcal{G}_1$ is the class of all LL(k) grammars for arbitrary k, etc. It is well known that for i=1,...,14 the languages which can be generated by grammars in $\mathcal{G}_i$ are deterministic context-free [1]. Moreover grammars G_1 and G_2 belong to $\mathcal{G}_i$ for every i=1,...,14.

So let us assume that for some i, where $1 \leq i \leq 14$, $\mathcal{G}_i$ is a family of grammars, i.e. there exists a grammar G such that $\mathcal{G}(G) = \mathcal{G}_i$. Hence G_1 and G_2 belong to $\mathcal{G}(G)$ and $L(G_1)$ and $L(G_2)$ belong to the grammatical family $\mathcal{L}(G)$. Since $\mathcal{L}(G)$ is closed under union, it follows that $L_1 \cup L_2$ is also in $\mathcal{L}(G)$ and hence there exists a grammar G_I which is an interpretation grammar of the grammar G such that $L_1 \cup L_2 = L(G_I)$. Hence G_I belongs to $\mathcal{G}(G)$ and thus to $\mathcal{G}_i$. Since $L_1 \cup L_2$ is not a deterministic context-free language [7] we obtain a contradiction. □

It is therefore reasonable to assume that for all the above mentioned grammar types there does not exist any natural way to define them in terms of productions.

We would be misleading the reader if we did not investigate the previously asked question from a

slightly different point of view: There are context-free grammars which generate LL(k) languages, but which themselves are not LL(k) grammars. Thus the family of LL(k) grammars is a proper subset of the family of all grammars that generate exactly the LL(k) languages. Similar situations occur for the other types of grammars listed above. Thus our claim that restrictions on productions and restrictions on derivations have very little connection would be only partially true if one could characterize, for example, all grammars that generate LL(k) languages for a given k in terms of a production type. However, a trivial change of the proof of theorem II.5 shows that, even from this modified point of view, there is only little connection between productions and derivations. We therefore list as a corollary:

Corollary: Each of the following classes of languages does not constitute a grammatical family, i.e., there is no grammar form F such that all interpretation grammars of F generate exactly any one of the following classes of languages:

1) LL(k) languages for arbitrary k
2) LR(k) languages for arbitrary k
3) LL(k) languages for any given k
4) LR(k) languages for any given k
5) Strong LL(k) languages for any given k
6) Simple LR(k) languages for any given k
7) Strict deterministic languages
8) Simple precedence languages
9) Uniquely invertible extended precedence languages
10) Uniquely invertible weak precedence languages
11) Simple mixed strategy precedence languages
12) Mixed strategy precedence languages
13) (1,1) bounded right context languages
14) Bounded right context languages.

Other classes of grammars and languages that have interesting parsing properties are the unambiguous grammars and languages of ambiguity degree k. The definition of unambiguity and ambiguity of degree k is based on the concept of derivations. Here too, we can easily show that there seems to be no way to "grasp" the concept of unambiguity solely in terms of productions.

Theorem II.6: Each of the following classes of grammars does not constitute a family of grammars:

1) Unambiguous context-free grammars
2) Ambiguous context-free grammars
3) Context-free grammars with ambiguity degree $\leq k$ for any given k
4) Context-free grammars with ambiguity degree $\geq k$ for any given k

Proof: to 1) Assume that there is a grammar G such $\mathcal{G}(G)$ equals the class of all unambiguous context-free grammars. Then the grammar $G=(V,\Sigma,P,S)$ has to be unambiguous since G is clearly an interpretation grammar of itself. Furthermore, $L(G)$ has to be an infinite language, since otherwise all grammars in $\mathcal{G}(G)$ and thus all unambiguous context-free grammars could generate only finite languages which clearly is a contradiction. Hence there exists a symbol $A \in (V-\Sigma)$ such that $S \overset{*}{\Rightarrow} uAz \overset{+}{\Rightarrow} uvAyz \overset{*}{\Rightarrow} uvxyz$ for some $u,v,x,y,z \in \Sigma^*$. We define an interpretation of G as follows:

$\mu(\alpha) = \{\alpha\}$ for every $\alpha \in V-\{A\}$

$\mu(A) = \{A,A'\}$ where $A' \notin V$.

Then $G_I = (V \cup \{A'\},\Sigma,\mu(P),S)$ is an interpretation of G. Hence G_I should be an unambiguous grammar. However G_I is a grammar with an infinite degree of ambiguity. The string uv^kxy^kz has 2^{k+1} leftmost

derivations, for every $k \geq 1$.

to 2) Assume that there is a grammar G such that $\mathcal{L}(G)$ equals the class of all ambiguous grammars. Then G has to be ambiguous. It is trivial to find an unambiguous interpretation of G and thus arrive at a contradiction.

to 3 and 4) Similar to cases 1) and 2). □

The class of grammars that generate unambiguous context-free languages is strictly larger than the class of unambiguous grammars. So again we have to address the question whether all grammars generating unambiguous context-free languages can be characterized in terms of productions. The answer is again "no".

Theorem II.7: Each of the following families of languages does not constitute a grammatical family:

1) Unambiguous context-free languages, and
2) inherently ambiguous context-free languages.

Proof: To 1) It is well known that there are inherently ambiguous context-free languages [11]. So if there exists a grammar G such that $\mathcal{L}(G)$ equals the set of all unambiguous context-free languages then $\mathcal{L}(G)$ contains all context-free languages since $\mathcal{L}(G)$ is closed under homomorphism and since every context-free language L_1 can be expressed as $h(L_2)$ with h a homomorphism and L_2 an unambiguous context-free language. Thus we obtain a contradiction.

To 2) Similar to proof of part two of Theorem II.6. □

III. THE INFLUENCE OF PRODUCTIONS ON PARSING

In [3] Bertsch has shown that so-called strict interpretations[3,5] of unambiguous grammar forms can be parsed in essentially the same time (up to a multiplicative constant c) as the underlying unambiguous form grammar. One of the conditions, which is quite restrictive and essential for a proof in [3] (cf. Theorem III.3), is missing in the definition of a strict interpretation in [3]. So the following question arises: "How restrictive are strict interpretations?" We will show that grammars given by strict interpretations of an unambiguous form grammar generate only unambiguous languages. Similarly, strict interpretations of an LR(k) grammar generate only LR(k) languages. This is not so in the LL case. Finally, we will prove that it is decidable for a given unambiguous grammar form whether a strict interpretation is unambiguous or not. It is also decidable for two "compatible" strict interpretations G_1 and G_2 of an unambiguous form grammar whether $L(G_1) \subseteq L(G_2)$, $L(G_1) = L(G_2)$, and whether $L(G_1) \cap L(G_2)$ is empty, finite, or infinite.

The formal framework used in [3] is that of x-categories and x-functors. The reader is referred to [2],[3],[12],[13],[17], and [20] for further details on these concepts. The definition of a strict interpretation given in [3] is as follows: An interpretation $I = (\mu, V_I, \Sigma_I, P_I, S_I)$ (of a grammar form $F = (\mathcal{V}, \mathcal{S}, V, \Sigma, P, S)$) is a strict interpretation if

$\mu(a)$ is a finite subset of $\mathcal{S}$ where $a \in \Sigma$.

We quote the following theorem from [3].

Theorem III.1: Let G_F be the form grammar of a context-free grammar form F. Then G_I is the grammar of a strict interpretation of F if and only if there exists a length-preserving x-functor $\phi: G_I \to G_F$.

One can immediately show that if $\mu(a) \cap \mu(b) \neq \emptyset$ for two symbols a and $b \in \Sigma$ (a case that is al-

lowed under the definition of a strict interpretation in [3]) then one cannot construct the length-preserving x-functor $\phi: G_I \to G_F$. The additional condition needed is one that requires $\mu(a) \cap \mu(b) = \emptyset$ for all $a, b \in \Sigma$. Thus we need to add this condition to the definition of a strict interpretation. (Definition III.2 was also used in [5].)

Definition III.2: An interpretation $I = (\mu, V_I, \Sigma_I, P_I, S_I)$ (of a grammar form $F = (V, S, V, \Sigma, P, S)$) is a strict interpretation if

1) $\mu(a)$ is a finite subset of S $\forall a \in \Sigma$ (length-preserving property) and
2) $\mu(a) \cap \mu(b) = \emptyset$ $\forall a, b \in \Sigma$ (disjoint-images property).

With this definition of a strict interpretation Theorem III.1 holds.

We would like to quote the following Theorem from [3]:

Theorem III.3: Let $\phi: G_1 \to G_2$ be a length-preserving functor, where G_1 and G_2 are context-free grammars and G_2 is unambiguous. Suppose there is an algorithm which will construct a parse for $w \in L(G_2)$ and reject $w \notin L(G_2)$ in less than $f(|w|)$ steps. Then there is a constant c and an algorithm which will accept $w \in L(G_1)$ and reject $w \notin L(G_1)$ in less than $cf(|w|)$ steps.

An obvious consequence, as stated in [3], is:

Corollary: Let G_F be an unambiguous form grammar of a grammar form F. Then strings in $L(G_I)$ for a strict interpretation G_I of G_F can be parsed in essentially the same time (up to a multiplicative constant c) as strings in $L(G_F)$.

One of the reasons for mentioning the above facts is that strict interpretations are more restricted than one might suspect. And as we will now show, it is the disjoint images property, and not so much the length-preserving property, that limits the applicability of Theorem III.3 and its Corollary. Most of the proofs in the remainder are only sketched, and the interested reader is referred to [14].

Theorem III.4: For every form grammar G_F of a grammar form F there exists a grammar form $\overline{F}$ such that $L(G_F)$ is equal to the class of all languages that can be generated by grammars given by length-preserving interpretations of $\overline{F}$. If G_F is unambiguous then $G_{\overline{F}}$ can be made unambiguous.

In order to show that the context-free languages can be generated by grammars given by length-preserving interpretations of an unambiguous grammar form, we state the following theorem:

Theorem III.5: There is an unambiguous form grammar G_F of a grammar form F such that $L(G_F)$ is equal to the class of context-free languages.

The proof of Theorem III.5 uses the following facts:

1) There is an unambiguous grammar G which generates the Dyck-language over two letters [6].
2) L(G) is infinite.
3) $L(G)$ is closed under homomorphism, inverse homomorphism, and intersection with regular sets.
4) Every context-free language L can be expressed as $L = h_1(h_2^{-1}(D) \cap R)$ with h_1 and h_2 being homomorphisms, R a regular set, and $D = L(G)$.

Combining Theorems III.4 and III.5 we obtain the following corollary:

Corollary: There is an unambiguous form grammar G_F such that the grammars given by length-preserving interpretations of G_F generate exactly the context-free languages.

Hence, the length-preserving property is not

very stringent and it does not limit the applicability of Theorem III.3. However, the case is quite different with respect to the disjoint-images property which limits the applicability of Theorem III.3. But this also has an advantage in that strict interpretations are more "structure preserving" than unrestricted interpretations.

Notation: Henceforth we will use the following notation:

$G^S(G_F) = \{G \mid G$ is a strict interpretation of $G_F\}$, and

$L^S(G_F) = \{L(G) \mid G \in G^S(G_F)\}$.

Theorem III.6: If G_F is an unambiguous form grammar with $L(G_F)$ infinite then there are strict interpretations of G_F that are ambiguous.

However, the existence of a length-preserving functor $\phi: G_I \to G_F$ for a strict interpretation G_I of an unambiguous grammar G_F tells us that, although there might be many distinct leftmost derivations for a string $w \in L(G_I)$, that the corresponding derivation trees look the same. They might differ only in their labels for the nonterminal nodes. Thus one suspects that strict interpretations of unambiguous form grammars generate only unambiguous languages. This is proven by the following theorem:

Theorem III.7: Let G_F be an unambiguous form grammar and let G be an arbitrary strict interpretation of G_F. Then there exists (effectively) an unambiguous grammar G' such that $L(G) = L(G')$.

The proof of Theorem III.7, which is based on Definitions III.8 and III.10, Lemma III.9, and Algorithm D, follows from Lemma III.11 and Lemma III.12.

Definition III.8: Let $G = (V,\Sigma,P,S)$ be a context-free grammar. Let $G_I = (V_I,\Sigma_I,P_I,S_I)$ be a strict interpretation grammar of G given by μ. A merge set for G_I is $M \subseteq \mu(B)$ such that $B \in V\text{-}\Sigma$, and such that $(\forall B' \in M)(\forall B'' \in V_I\text{-}\Sigma_I)$

$[(B' \in M$ and $B'' \in M)$ iff $(\exists \alpha \in V_I^*)$

$[B' \to \alpha \in P_I$ and $B'' \to \alpha \in P_I]]$.

Lemma III.9: Let $G = (V,\Sigma,P,S)$ be a context-free grammar. Let $G_I = (V_I,\Sigma_I,P_I,S_I)$ be a strict interpretation given by μ. If G is unambiguous but G_I is ambiguous, then there exists a merge set for G_I.

Definition III.10: Let $G \in G^S(G_F)$ under μ. Then $M(G) = \{M \mid M$ is a merge set for G under $\mu\}$.

Algorithm D: Let $G_F = (V_F,\Sigma_F,P_F,S_F)$ be an unambiguous context-free grammar. Let $G = (V,\Sigma,P,S) \in G^S(G_F)$ via the mapping μ.

Output: A CFG G' such that $L(G') = L(G)$ and G' is unambiguous.

Procedure:

1) Construct the merge sets $M(G)$.

2) If G is ambiguous then, by Lemma III.9, $|M(G)| \geq 1$. Thus if $|M(G)| = 0$, let $G' = G$ and halt.

3) Let $B = \{B_X \mid X \subseteq M$ and $M \in M(G)\}$ be a set of new symbols mutually disjoint from V.
Let $\overline{M} = \{B \mid B \in M$ and $M \in M(G)\}$.
Let $V' = V\text{-}\overline{M} \cup B$ if $S \notin \overline{M}$
or $= V\text{-}\overline{M} \cup B \cup \{S\}$ if $S \in \overline{M}$.

4) Let $P_1 = P - ((\overline{M} \times V^*) \cup (V \times V^*\overline{M}V^*))$.

5) Let $P_2 = \{B_{X_0} \to \alpha_0 B_{X_1} \alpha_1 B_{X_2} \alpha_2 \cdots \alpha_m \mid B_0 \to \alpha_0 B_1 \alpha_1 B_2 \alpha_2 \cdots \alpha_m \in P$, $(\forall i)_{0 \leq i \leq m}(B_i \in \overline{M}$ and $\alpha_i \in (V\text{-}\overline{M})^*$ and $B_i \in X_i)$, and $X_0 = \{B_0\}\}$.

6) Let $P_3 = \{B_X \to \alpha \mid \exists M \in M(G)$ such that $X \subseteq M$ and $X = \{B_0 \mid B_{\{B_0\}} \to \alpha \in P_2\}\}$.

7) Let $P_4 = \{A \to \alpha_0 B_{X_1} \alpha_1 B_{X_2} \alpha_2 \ldots \alpha_m |$
$A \to \alpha_0 B_1 \alpha_1 B_2 \alpha_2 \ldots \alpha_m \in P,$
$(\forall i)_{0 \le i \le m} \alpha_i \in (V-\bar{M})^*,$
$(\forall i)_{1 \le i \le m} (B_i \in \bar{M}$ and $B_i \in X_i)$, and
$A \notin \bar{M}\}$.

8) Let $P_5 = \{S \to \alpha_0 B_{X_1} \alpha_1 \ldots B_{X_m} \alpha_m |$
$S \to \alpha_0 B_1 \alpha_1 \ldots B_m \alpha_m \in P,$
$(\forall i)_{0 \le i \le m} \alpha_i \in (V-\bar{M})^*,$
$(\forall i)_{1 \le i \le m} (B_i \in M_i$ and $B_i \in X_i)\}$.

9) Let $P' = P_1 \cup P_3 \cup P_4 (\cup P_5$ if $S \in \bar{M})$.
Let $G' = (V', \Sigma, P', S)$ and halt.

In the algorithm above, nonterminal symbols of G that may be a source of ambiguity, i.e. symbols in M, are eliminated. They are replaced by new nonterminal symbols based on subsets of the merge sets. B_X generates (in P') exactly that which the nonterminal symbols in X generate in common (in P). This leads to the following lemma:

Lemma III.11: The grammar produced as output by Algorithm D is unambiguous.

Lemma III.12: Let G be the input to Algorithm D and G' be the output of the algorithm. Then $L(G) = L(G')$.

Corollary (to Theorem III.7): If G_F is an unambiguous form grammar then $\mathcal{L}^S(G_F)$ contains only unambiguous languages.

There are context-free languages that are inherently ambiguous. Hence we obtain:

Theorem III.13: There is no unambiguous form grammar G_F such that the grammars given by strict interpretations of G_F generate exactly the context-free languages.

The technique used in the proof of Theorem III.7 can also be used to prove the following theorem.

Theorem III.14: If G_F is an LR(k) form grammar then $\mathcal{L}^S(G_F)$ contains only LR(k) languages. However, a similar statement for the LL(k) case is not true. Not only are there unambiguous LL(k) grammars whose strict interpretation grammars can be unambiguous and non-LL(k), there are unambiguous LL(k) grammars whose strict interpretations can yield grammars which generate non-LL languages.

Summarizing the first part of this section we can state that strict interpretations substantially limit the applicability of Theorem III.3. However, strict interpretations have an advantage over unrestricted interpretations since they are more structure preserving.

We will conclude this section with a few decidability results.

Theorem III.15: It is decidable whether a strict interpretation of an unambiguous grammar form is unambiguous.

The proof of this last theorem is based on the observation that for a strict interpretation G in reduced form one can establish a number k such that G is ambiguous if and only if there exist sentential forms uAv and $uA'v$ with $|uAv| = |uA'v| \le k$ such that A and A' are members of the same merge set.

Theorem III.15 can be used to prove the following decidability results for "compatible" strict interpretations of an unambiguous grammar form.

Definition III.16: Two strict interpretations $I_1 = (\mu_1, V_1, \Sigma_1, P_1, S_1)$ and $I_2 = (\mu_2, V_2, \Sigma_2, P_2, S_2)$ of a given grammar form are compatible if

$(\mu_1(a) \cup \mu_2(a)) \cap (\mu_2(b) \cup \mu_2(b)) = \emptyset$

for any two elements a, b in Σ.

Theorem III.17: It is decidable for any two compatible strict interpretations G_1 and G_2 of an unambiguous form grammar whether

1) $L(G_1) \subseteq L(G_2)$,
2) $L(G_1) = L(G_2)$,
3) $L(G_1) \cap L(G_2)$ is empty, finite or infinite.

IV. COMPARISON OF STRICT INTERPRETATIONS WITH UNRESTRICTED INTERPRETATIONS

In section III we have seen that there exists an unambiguous form grammar G_F such that $L(G_F)$ is equal to the family of context-free languages (Thoerem III.3), but that there is no unambiguous form grammar G_F such that $L^S(G_F)$ equals the class of context-free languages (Theorem III.13). So we want to investigate the question of whether grammar forms with strict interpretation are as "powerful" as grammar forms with unrestricted interpretations if we do not consider the ambiguity or unambiguity of the form grammar. First we obtain:

Theorem IV.1: There are grammar forms F with form grammar G_F such that there does not exist any grammar form $\overline{F}$ with form grammar $G_{\overline{F}}$ such that $G^S(G_{\overline{F}}) = G(G_F)$.

So with respect to the interpretation grammars strict interpretations are not as powerful as unrestricted interpretations. The situation is different though, if one looks at the classes of languages which can be generated by the interpretation grammars.

Theorem IV.2: For every form grammar G_F there exists a form grammar $G_{\overline{F}}$ such that $L^S(G_{\overline{F}}) = L(G_F)$.

V. CONCLUSION

We have seen that grammar forms with unrestricted interpretations do not provide a characterization of many grammar types that are used in parsing.

Strict interpretations of unambiguous grammar forms are structure preserving. They preserve unambiguity and LR(k)-ness of the generated languages. They also provide us with some interesting decidability results. There is a relation between the parsing time for the interpretation grammars and the parsing time for the underlying unambiguous form grammar.

Several questions arise naturally from these observations:

1) Is there a "universal" parser for all LL(k) languages?
2) Is there a "universal" parser for all LR(k) languages?
3) If the answer to questions 1) and 2) is "no", are there "universal" parsers for "reasonably" large subclasses of the LR(k) languages, LL(k) languages, and other classes of languages?
4) Is there a master grammar which we know to be amenable to a fast parse technique but which generates a much larger class of grammars which have not been suspected before of having a fast parse technique?

The results given in this paper constitute an initial step in this direction and we encourage further research along these lines.

ACKNOWLEDGEMENT: The authors are grateful to S. Ginsburg for stimulating their interest in grammar forms.

REFERENCES

[1] Aho, Alfred V. and Jeffrey D. Ullman. The Theory of Parsing, Translation, and Compiling, Vol. I and II, Prentice Hall, Englewood Cliffs,

New Jersey, 1972.

[2] Bertsch, Eberhard. "An Observation on Relative Parsing Time," JACM, Vol. 22, No. 4, pp. 493-498, 1975.

[3] Bertsch, E. "Mappings between Context-Free Derivation Systems," Lect. Notes in Comp. Sci., Vol. 2, pp. 278-283, 1973.

[4] Cremers, A. B. and S. Ginsburg. "Context-Free Grammar Forms," JCSS, Vol. 11, pp. 86-117, 1975.

[5] Cremers, A. B., S. Ginsburg, and E. H. Spanier. "The Structure of Context-Free Grammatical Families," submitted for publication.

[6] Ginsburg, S. The Mathematical Theory of Context-Free Languages, McGraw-Hill, New York, 1966.

[7] Ginsburg, S. and S. A. Greibach. "Deterministic Context Free Languages," Inform. and Control, Vol. 9, No. 6, pp. 620-648, 1966.

[8] Ginsburg, S. and M. Harrison. "Bracketed Context-Free Languages," JCSS, Vol. 1, pp. 1-23, 1967.

[9] Greibach, S. "The Hardest Context-Free Language," SIAM J. Computing, Vol. 2, pp. 304-310, 1973.

[10] Harrison, M. A. and I. M. Havel. "Strict Deterministic Grammars," JCSS, Vol. 7, 1973.

[11] Hopcroft, John E. and Jeffrey D. Ullman. Formal Languages and Their Relation to Automata, Addison-Wesley, Reading, Mass., 1969.

[12] Hotz, G. "Eindeutigkeit und Mehrdeutigkeit Formaler Sprachen," Elektr. Inform. und Kybernetik, 1966.

[13] Hotz, G. "Homomorphie und Aequivalenz Formaler Sprachen," JSNM, Vol. 6, Birkhäuser Verlag, Basel, 1967.

[14] Leong, B. L. and D. Wotschke. "Productions, Derivations, and Parsing," Tech. Rep., in preparation.

[15] Knuth, D. "A Characterization of Parenthesis Grammars," Inform. and Control, Vol. 11, pp. 269-289, 1967.

[16] McNaughton, R. "Parenthesis Grammars," JACM, Vol. 14, pp. 490-500, 1967.

[17] Mitchell, B. Theory of Categories, Academic Press, New York, 1965.

[18] Paull, M. and M. Unger. "Structural Equivalence of Context-Free Grammars," JCSS, Vol. 2, pp. 427-463, 1965.

[19] Salomaa, A. Formal Languages, Academic Press, New York and London, 1973.

[20] Schnorr, C. P. "Transformational Classes of Grammars," Inform. and Control, Vol. 14, pp. 252-277, 1969.