A Complexity Theory of Grammar Problems

H. B. Hunt, III[+]

Aiken Computation Laboratory
Harvard University
Cambridge, Massachusetts 02138

1. Introduction

The close relationship between programming language syntax, context-free grammars (abbreviated cfgs), parsing, and compiling is well-known and is extensively discussed in [1]. Unfortunately, many of the problems about programming languages, one might wish to solve, are equivalent to undecidable grammar problems. Two especially important such problems are

(1) the emptiness of intersection problem, i.e. determining if the intersection of the languages generated by a pair of grammars is empty, and
(2) the grammar class membership problem, i.e. determining for a fixed class of grammars Γ and a grammar G, if G is an element of Γ.

Here, we investigate two recently proposed ideas for circumventing the undecidability of such problems.

The separability concept was introduced in [2] in an attempt to ameliorate the undecidability of such problems as the emptiness of intersection problem. Intuitively, the idea of separability is to replace arbitrary context-free languages (cfls) by approximating languages for which such problems are decidable. The inherent limitations of the separability concept are discussed. We show that for many different families of separating languages F, it is undecidable if some language in F separates the languages generated by two arbitrary cfgs. We also show that there are fixed cfls L_0 such that for any family of languages F containing L_0, it is undecidable if some language in F separates the languages generated by two arbitrary cfgs. These cfls include $\{w\#w^{rev} \mid w\in\{0,1\}^*\}$, $\{ww^{rev} \mid w\in\{0,1\}^*\}$, and all Dyck languages over two or more letters.

The decidability of a variety of problems related to problem (2) above is considered. We show that for many classes Γ of cfgs and many binary relations ρ on the cfgs, all predicates of the form, "for an arbitrary cfg G, $G\rho H$ for some H in Γ", are undecidable. Thus for example, letting Γ denote the class of strong LL, LL, strong LC, LC, SLR, LALR, LR, LR regular, or unambiguous cfgs, it is undecidable if an arbitrary cfg is, is equivalent to, is structurally equivalent to, is covered by, or covers, an element of Γ. The problem of deciding for a class Δ of cfgs, a class T of grammatical transformations, and an arbitrary cfg G, if some transformation t in T maps G to an element of Δ, is also considered. Thus for example, if each t in T is language preserving, i.e. for all cfgs G, $L(t(G)) = L(G)$, and the identity transformation is in T, then for an arbitrary cfg G, it is undecidable if there exists t in T such that t(G) is a strong LL, LL, strong LC, LC, SLR, LALR, LR, LR regular, or unambiguous cfg. Our results apply to many classes of transformations that do not contain the identity transformation as well.

We list several definitions and observations used in the remainder of this paper. We assume that the reader is familiar with the basic definitions of and results concerning cfgs, cfls, the strong LL, SLR, and BRC grammars, Turing machines (Tms), and Turing machine computations, otherwise see [1] or [3].

<u>Definition 1.1</u>: A language L over a finite alphabet Σ is said to be

(i) a <u>definite event</u> if and only if $L=A\cdot\Sigma^*\cup B$, where $A,B\subseteq\Sigma^*$ are finite sets;

and is said to be

[+] This research was supported in part by U.S. Army Contract No. DA-31-124-ARO-D-462 and by NSF Grant No. DCR75-22505.

(ii) a reverse definite event if and only if $L=\Sigma^{*}\cdot A \cup B$, where $A,B\subseteq\Sigma^{*}$ are finite sets.

We note that all definite and reverse definite events are regular sets. Moreover, the classes of definite and reverse definite events are closed under complementation.

The next two definitions formalize the separability concept.

Definition 1.2: Let L_1, L_2, and S be languages over a finite alphabet Σ. We say that S separates L_1 and L_2, or equivalently L_1 and L_2 are separated by S if and only if
$L_1\subseteq S$ and $L_2\cap S=\phi$, or $L_1\cap S=\phi$ and $L_2\subseteq S$.

Definition 1.3: Let F be any family of languages. Two languages L_1 and L_2 are said to be

(i) F separable if and only if there exists a language $S\in F$ such that S separates L_1 and L_2;

and are said to be

(ii) F H-separable if and only if there exist two disjoint languages S_1 and $S_2\in F$ such that $L_1\subseteq S_1$ and $L_2\subseteq S_2$. ‡

Clearly if two languages are F H-separable, then they are F separable. However, F separability does not in general imply F H-separability. If F is closed under complementation, then F separability and F H-separability are the same. Thus if F is the class of definite or reverse definite events over some finite alphabet Σ, then two languages L_1 and L_2 are F separable if and only if they are F H-separable.

2. The Basic Construction

In this section the basic technical construction needed to develop our complexity theory of grammar problems is presented. Unlike previous work, we do not prove the undecidability of a problem Π by effectively reducing the halting problem for Tms, or some equivalent problem, to Π. Rather, we prove the undecidability of Π by efficiently reducing the membership problem for Turing machines that always halt to it. This shows that Π is of nonrecursive computational complexity. For the problems considered here, this suffices to show that they are undecidable.

Let M be any deterministic Tm, with tape alphabet T and state set S, that always halts on the right end of its tape. Then M is O(T(n)) time-bounded for some strictly inreasing recursive function T(n). Given an input string x to M, two cfgs $G_1[M,x]$ and $G_2[M,x]$ can be constructed in linear time on a deterministic multi-tape Tm such that

$L(G_1[M,x])=\#\cdot\{y\cdot\#\cdot z^{rev}\cdot\# \mid y, z$ are i.d.'s of M and $y\vdash_M z\}^{*}\cdot\$$, and

$L(G_2[M,x])=\#\cdot x_1\cdot\{\#\cdot y^{rev}\cdot\#\cdot y \mid y$ is an i.d. of M$\}^{*}\cdot\{\#\cdot z^{r}\cdot\# \mid z$ is an accepting i.d. of M$\}\cdot\$$, where x_1 is an initial i.d. of M on x.

Let $\Sigma=T\cup(S\times T)\cup\{\#,\$\}$. We assume that T, $(S\times T)$, and $\{\#,\$\}$ are mutually disjoint.

For sufficiently fast strictly increasing functions T(n), no pair of words in $L(G_1[M,x])$ and $L(G_2[M,x])$ have common prefixes of length $\geq c\cdot[T(|x|)]^2$ for some positive integer c depending only upon M not x. Moreover, $G_1[M,x]$ and $G_2[M,x]$ are strong LL, SLR, and BRC grammars.

Proposition 2.1: Let M, T(n), c, Σ, and x be as described above. Let F be any class of languages over Σ containing all definite events over Σ. Let $k=c\cdot[T(|x|)]^2$.

Then in time $\leq c_1\cdot|x|$, cfgs G_1, G_2, and G_3 each of size $\leq c_2+|x|$, where c_1 and c_2 are constants depending only upon M not x, can be constructed by a deterministic multi-tape Tm such that the following are equivalent:

(1) M accepts x;
(2) $L(G_1)\cap L(G_2)\neq\phi$;
(3) $L(G_1)$ and $L(G_2)$ are not definite event separable;
(4) $L(G_1)$ and $L(G_2)$ are not F separable;
(5) $L(G_1)$ and $L(G_2)$ are not F H-separable;
(6) G_3 is ambiguous;
(7) G_3 is inherently ambiguous;
(8) G_3 is not a strong LL(k) grammar;
(9) G_3 is not an SLR(k) grammar; and
(10) G_3 is not a BRC(k,k) grammar.

Proof: G_1 and G_2 are equal to $G_1[M,x]$ and $G_2[M,x]$ respectively. G_3 is the cfg whose productions consist of

(a) all productions of G_1 and G_2;
(b) $S\rightarrow AS_1\$S_3 \mid B¢S_2\$S_4$, where S_1 and S_2 are the start symbols of G_1 and G_2, respectively;
(c) $A\rightarrow ¢¢$;
(d) $B\rightarrow ¢$;
(e) $S_3\rightarrow aTbcU$;
(f) $T\rightarrow aTb \mid ab$;
(g) $U\rightarrow cU \mid c$;
(h) $S_4\rightarrow aVbWc$;
(i) $V\rightarrow aV \mid a$; and
(j) $W\rightarrow bWc \mid bc$.

We assume that $\{S,A,S_3,B,S_4,T,U,V,W\}$ is disjoint from the union of the nonterminal alphabets of G_1 and G_2.

‡H-separability is named after Hausdorf separability in topology.

If M accepts x then there exists a string $w \in L(G) \cap L(G_2)$. Thus $L(G_1)$ and $L(G_2)$ are not F or F H-separable for any family of languages F. Moreover,

$$L(G_3) \cap ¢¢\cdot w\cdot \$\cdot \Sigma* = ¢¢\cdot w\cdot \$\cdot$$
$$\{a^n b^n c^m \mid n,m \geq 2\} \cup \{a^n b^m c^m \mid n,m \geq 2\}].$$

Since the unambiguous cfls are closed under intersection with regular sets and $L(G_3) \cap ¢\cdot ¢\cdot w\cdot \$\cdot \Sigma*$ is an inherently ambiguous cfl, G_3 is an inherently ambiguous cfg. Hence a fortiori, G_3 is ambiguous and is not a strong $LL(\ell)$, $SLR(\ell)$, or $BRC(\ell,\mu)$ grammar for any choice of ℓ and μ. If M does not accept x, then $L(G_1) \cap L(G_2) = \emptyset$ and no pair of words in $L(G_1)$ and $L(G_2)$ have a common prefix of size k. Thus $L(G_1)$ and $L(G_2)$ are definite event separable, F separable, and F H-separable. Moreover by inspection of the productions of G_3, G_3 is a strong LL(k), an SLR(k), and a BRC(k,k) grammar.

In Sections 3 and 4, Proposition 2.1 is used to derive most of our grammar complexity results. More details of the constructions of the cfgs G_1, G_2, and G_3 can be found in [4].

Finally, our undecidability proofs make use of the following two properties of recursive sets and Tms that always halt.

Property 2.2: Every deterministic Tm that always halts makes $\leq T(n)$ moves for some strictly increasing recursive function T(n) and for all inputs of size n.

Property 2.3 For all recursive functions f, there is a recursive set S_f such that any Tm that recognizes S_f requires time > f(n) for infinitely many n.
A proof of Property 2.3 can be found in [5].

3. Separability

Separability was introduced in [2] in an attempt to ameliorate the undecidability of such problems as the emptiness of intersection problem. The idea of separability is to approximate arbitrary cfls by languages for which such problems are decidable. Here the limitations of the separability concept are considered. We consider the question – "For what families F of languages is it decidable if an arbitrary pair of cfls are F or F H-separable?" – We show that this question is undecidable for many, if not most, interesting families of languages F.

One obvious candidate for such an F is the family of regular sets.

Definition 3.1[2]: Two languages L_1 and L_2 are said to be regular separable if there exist disjoint regular sets R_1 and R_2 such that $L_1 \subseteq R_1$ and $L_2 \subseteq R_2$.[1]

Results in [2] and [8] show that regular separability is undecidable for the cfls. Another obvious candidate for F is the family of cfls. But letting F equal the cfls, two cfls L_1 and L_2 are F or F H-separable if and only if $L_1 \cap L_2 = \emptyset$. Thus again F and F H-separability are undecidable. We show that both the undecidability of regular separability and the undecidability of emptiness of intersection of the cfls follow from a general separability undecidability theorem, Theorem 3.2 below.

Theorem 3.2: Let $\hat{\Sigma}$ be any finite alphabet of cardinality greater than one. Let F be any family of languages that contains all definite or reverse definite events over $\hat{\Sigma}$. Then it is undecidable if the languages generated by two arbitrary cfgs with terminal alphabets equal to $\hat{\Sigma}$ are

(i) F separable, or (ii) F H-separable.

Proof: We only prove the theorem for the case, when F contains all definite events over $\hat{\Sigma}$. The proof for the case, when F contains all reverse definite events over $\hat{\Sigma}$, is similar and is left to the reader. The proof consists in showing that the decidability of F (or F H-separability), for any F containing all definite events over $\hat{\Sigma}$, contradicts Property 2.3.

Suppose for some such F that F (or F H-)separability is decidable for the cfgs. Then since the set of cfgs with terminal alphabets equal to $\hat{\Sigma}$ is a recursive set, by Proposition 2.2 there exists a strictly increasing recursive function f(n) that bounds the time required to decide F (or F H-separability) on a deterministic Tm. Let M be any O(T(n)) time-bounded deterministic Tm with T(n) a strictly increasing recursive function. We claim that L(M) is recognizable by a cn + f(cn) time-bounded deterministic Tm $\mathcal{M}$, where c is a constant depending only upon M.

Intuitively given input x to $\mathcal{M}$, we would like $\mathcal{M}$ to construct the cfgs $G_1 = G_1[M,x]$ and $G_2 = G_2[M,x]$ of Proposition 2.1 and then to test if $L(G_1)$ and $L(G_2)$ are F (or F H-)separable. However, G_1 and G_2 of Proposition 2.1 can not be used directly because their terminal alphabets need not equal $\hat{\Sigma}$. A straight-forward encoding can be used to construct cfgs $\hat{G}_1$ and $\hat{G}_2$ with terminal alphabets equal to $\hat{\Sigma}$ such that the time and space required for their construction is still linear in |x| and such that conclusions (1) through (5) of Proposition 2.1 hold with $\hat{G}_1$ and $\hat{G}_2$ substituted for G_1 and G_2, respectively.

1. In our terminology, two languages are regular separable if and only if they are F H-separable, with F equal to the family of regular sets.

The Tm $\mathcal{M}$ operates as follows.

1. Given input x, $\mathcal{M}$ constructs $\hat{G}_1$ and $\hat{G}_2$.
2. $\mathcal{M}$ tests if $L(\hat{G}_1)$ and $L(\hat{G}_2)$ are F (or F H-)separable. If so $\mathcal{M}$ rejects x. If not $\mathcal{M}$ accepts x. Step 1 requires at most cn time and step 2 requires at most f(cn) time, for some constant c depending only upon M, not x or $n = |x|$. If $x \notin L(M)$, then $L(\hat{G}_1)$ and $L(\hat{G}_2)$ are definite event, F, and F H-separable. Thus $\mathcal{M}$ rejects x. If $x \in L(M)$, then $L(\hat{G}_1) \cap L(\hat{G}_2) \neq \phi$, and $L(\hat{G}_1)$ and $L(\hat{G}_2)$ are not F or F H-separable for any family of languages F. Thus $\mathcal{M}$ accepts x.

Finally for all positive integers a, the recursive function $F(n) = n^2 + f(n^2)$ is strictly greater than an + f(an), for all but finitely many values of n. Thus if F (or F H-)separability is decidable for the cfgs, the every recursive set is accepted by some Tm that is F(n) time-bounded, for all but finitely many values of n. This contradicts Proposition 2.3

Theorem 3.2 shows how little is required for the undecidability of F or F H-separability. The following corollary illustrates its power and applicability.

Theorem 3.3: Let $\hat{\Sigma}$ be any finite alphabet of cardinality greater than one. The following classes F of languages over $\hat{\Sigma}$ satisfy the conditions of Theorem 3.2:

(1) definite events;
(2) reverse definite events;
(3) noncounting events;
(4) regular sets;
(5) for all $k \geq 1$, the LL(k) languages;
(6) LL languages;
(7) deterministic cfls;
(8) unambiguous cfls;
(9) cfls;
(10) linear cfls;
(11) context sensitive languages; and
(12) recursively enumerable sets.

Thus letting F denote any of the above classes of cfls, it is undecidable if the languages generated by two arbitrary cfgs with terminal alphabet equal to $\hat{\Sigma}$ are F or F H-separable.

Theorem 3.3 follows immediately from Theorem 3.2 and known properties of language classes (1) through (12). (Definitions of these language classes can be found in [9](1-3), [1](4-10), and [3](11-12).) A slight modification of the proofs of Proposition 2.1 and Theorem 3.2 shows that the F and F H-separability problems for the cfgs are also undecidable for F equal to the s-languages in [10], the strict and real-time strict deterministic languages in [11], and the LR(0) languages defined in [1]. Finally since the cfgs G_1 and G_2 in the proof of Theorem 3.2 are strong LL, SLR, and BRC grammars, many of the conclusions of Theorem 3.2 hold for any class of grammars containing the intersection of these three grammar classes. Thus for example letting F denote the definite events, noncounting events or the regular sets, it is undecidable if the languages generated by two arbitrary strong LL, SLR, LALR, LR, or BRC grammars are F or F H-separable.

We have already shown that the F and F H-separability problems for the cfgs are undecidable for all families of languages that contain sufficiently many regular sets. Next a second sufficient condition on a family of languages F for the undecidability of F separability is presented. We show that there are fixed cfls L_0 such that any family of languages F containing L_0, it is undecidable if some language in F separates two arbitrary cfls. Before presenting this result, we need several definitions from [12].

Definition 3.4: (i) C is the class of all cfls L_0 for which it is decidable if the language generated by an arbitrary cfg G is a subset of L_0.
(ii) I is the class of all cfls L_0 for which it is decidable if the intersection of the language generated by an arbitrary cfg G and L_0 is empty.

Definition 3.5: A cfl L is said to be forbidden if for all families of languages F such that $L \in F$, it is undecidable if the languages generated by an arbitrary pair of cfgs are F separable.

Definition 3.6: Let L be a cfl. Let Σ be the smallest finite alphabet such that $L \subseteq \Sigma^*$. Then $\overline{L} = \Sigma^* - L$.

Theorem 3.7: There exist forbidden cfls. In particular if a cfl L is not an element of I, then L is forbidden. Similarly if both L and $\overline{L}$ are cfls, and L is not an element of C, then L is forbidden.
Proof: A proof that there exist cfls that are not elements of I can be found in [12]. We only show that if a cfl L is not an element of I, then L is forbidden. Let $L \notin I$. Let F be any family of languages containing L. Since L is a cfl, there exists a fixed cfg G_0 such that $L = L(G_0)$. For an arbitrary cfg G, L(G) and $\overline{L} = \overline{L}(G_0)$ are F separable if and only if $L(G) \cap L = \phi$. Thus since $L \notin I$, F separability is undecidable.

In [12] we show that $\{ww^{rev} | w \in \{0,1\}^*\}$, $\{w\#w^{rev} | w \in \{0,1\}^*\}$, and all Dyck languages over two or more letters are forbidden. The language $\{w\#w^{rev} | w \in \{0,1\}^*\}$ is especially interesting because it is a non-regular s-language. Theorem 3.7 and results in [12] can be used to show that F separability is undecidable for a variety of families F of languages that do not contain the definite or reverse definite events, e.g. the parenthesis languages [13], the Dyck languages, the non-regular s-languages, etc.

4. Complexity of Relations

Recently in [4], [14], and [15] a complexity theory for certain types of grammar problems was presented. these problems dealt primarily with the complexity of grammar class membership problems. Here, these results are extended to a variety of other types of grammar problems. We show that for many grammar classes Γ, and for many binary relations ρ on the cfgs, all predicates of the form, "For an arbitrary cfg G, there exists a cfg H in Γ such that $G\rho H$.", are undecidable. Thus for example, it is undecidable if an arbitrary cfg G is, is equivalent to, is structurally equivalent to, or is covered by, a strong LL, LL, strong LC, LC, SLR, LALR, or LR grammar.

In what follows $\hat{\Sigma}$ is any arbitrary finite alphabet of cardinality greater than one; and $\mathscr{C}$ equals the intersection of the classes of strong LL, SLR, and BRC grammars with terminal alphabets equal to $\hat{\Sigma}$.

Theorem 4.1: Let ρ be any binary relation on the cfgs such that

(i) for all G in $\mathscr{C}$, $G\rho H$ for some H in $\mathscr{C}$, and

(ii) for all inherently ambiguous cfgs G, $G\rho H$ implies that H is inherently ambiguous.

Let Γ be any class of cfgs such that

(iii) $\mathscr{C} \subseteq \Gamma$; and

(iv) Γ contains no inherently ambiguous cfgs.

Then for arbitrary cfg G with terminal alphabet equal to $\hat{\Sigma}$, it is undecidable if there exists a cfg H in Γ for which $G\rho H$.

Proof: The proof closely follows that of Theorem 3.2. Let $\mathscr{P}$ denote the predicate, "For an arbitrary cfg G with terminal alphabet equal to $\hat{\Sigma}$, there exists a cfg H in Γ for which $G\rho H$.". Suppose for some relation ρ, satisfying conditions (i) and (ii), and for some grammar class Γ, satisfying conditions (iii) and (iv), that the predicate $\mathscr{P}$ is decidable. Then by Proposition 2.2 there exists a strictly increasing recursive function f(n) that bounds the time required to decide $\mathscr{P}$ on a deterministic Tm. Let M be any O(T(n)) time-bounded deterministic Tm with T(n) a strictly increasing recursive function. We claim that L(M) is recognizable by a cn+f(cn) time-bounded deterministic Tm $\mathscr{M}$, where c is a constant depending only upon M.

Intuitively given input x to $\mathscr{M}$, we would like $\mathscr{M}$ to construct the cfg G_3 of Proposition 2.1 and then to test if there exists a cfg H in Γ for which $G_3\rho H$. However, G_3 of Proposition 2.1 cannot be used directly because its terminal alphabet need not equal $\hat{\Sigma}$. A straight-forward encoding can be used to construct a cfg $\hat{G}_3$ with terminal alphabet equal to $\hat{\Sigma}$ such that the time and space required to construct $\hat{G}_3$ is still linear in $|x|$ and such that conclusions (6) through (10) of Proposition 2.1 hold for $\hat{G}_3$ substituted for G_3.

The Tm $\mathscr{M}$ operates as follows.

1. Given input x, $\mathscr{M}$ constructs $\hat{G}_3$.
2. $\mathscr{M}$ tests if there exists a cfg H in Γ for which $\hat{G}_3\rho H$. If so $\mathscr{M}$ rejects x. If not $\mathscr{M}$ accepts x. Step 1 requires at most cn time and step 2 requires at most f(cn) time, for some constant c depending only upon M not x or $n=|x|$. If $x \notin L(M)$, then $\hat{G}_3$ is an element of $\mathscr{C}$ and by (i) and (iii) there exists a cfg H in Γ for which $\hat{G}_3\rho H$. Thus $\mathscr{M}$ rejects x. If $x \in L(M)$, then $L(\hat{G}_3)$ is inherently ambiguous and by (ii) and (iv) there does not exist a cfg H in Γ for which $\hat{G}_3\rho H$. Thus $\mathscr{M}$ accepts x.

Finally for all positive integers c, the recursive function $F(n)=n^2+f(n^2)$ is strictly greater than cn+f(cn), for all but finitely many values of n. Thus if $\mathscr{P}$ is decidable for the cfgs, then every recursive set is accepted by some Tm that is F(n) time-bounded, for all but finitely many values of n. This contradicts Proposition 2.3.

The meaning, power, and applicability of Theorem 4.1 are best illustrated by examples.

Theorem 4.2: Let ρ be any reflexive binary relation on the cfgs such that for all cfgs G, H, $G\rho H$ implies $L(G)=L(H)$. Let Γ be any class of cfgs such that

(i) $\mathscr{C} \subseteq \Gamma$; and

(ii) Γ contains no inherently ambiguous cfgs.

Then for arbitrary cfg G with terminal alphabet equal to $\hat{\Sigma}$, it is undecidable if there exists a cfg H in Γ for which $G\rho H$.

Proof: The relation ρ satisfies conditions (i) and (ii) of Theorem 4.2.

Theorem 4.3: Let Γ be any class of cfgs such that

(i) $\mathscr{C} \subseteq \Gamma$; and

(ii) Γ contains no inherently ambiguous cfgs.

Then for an arbitrary cfg G with terminal alphabet equal to $\hat{\Sigma}$, it is undecidable if

G is, is equivalent to, is structurally equivalent to, covers, or is covered by, an element of Γ.

Proof: The definitions of structural equivalence and (grammatical) covering can be found in [1]. Let ρ be any of the following binary relations on the cfgs:

(1) GρH if and only if G=H;
(2) GρH if and only if L(G)=L(H);
(3) GρH if and only if G is structurally equivalent to H;
(4) GρH if and only if G covers H; and
(5) GρH if and only if G is covered by H.

Then ρ is reflexive and GρH implies L(G)=L(H). Thus each such relation ρ satisfies the conditions of Theorem 4.2, and Theorem 4.3 follows directly.

Corollary 4.4: Let Γ be any of the following grammar classes:

(1) BRC grammars;
(2) strong LL grammars;
(3) LL grammars;
(4) strong LC grammars;
(5) LC grammars;
(6) ELC grammars;
(7) SLR grammars;
(8) LALR grammars;
(9) LR grammars;
(10) LR regular grammars;
(11) unambiguous grammars;
(12) basic SPM parsable grammars;
(13) full SPM parsable grammars;
(14) FSPA grammars; and
(15) RPP grammars.

Then for an arbitrary cfg G with terminal alphabet equal to $\hat{\Sigma}$, it is undecidable if G is, is equivalent to, is structurally equivalent to, covers, or is covered by, an element of Γ.

Proof: Definitions of the above grammar classes can be found in [1](1-5,7-9,11), [16](6), [2](10), [17](12-13), and [18](14-15). The corollary follows immediately from Theorem 4.3.

Definition 4.5: (i) Any function t with domain and range equal to the family of cfgs (the family of cfgs with terminal alphabets equal to $\hat{\Sigma}$) is called a grammatical transformation on the cfgs (on the cfgs with terminal alphabets equal to $\hat{\Sigma}$.)
(ii) A grammatical transformation t is said to be language preserving if and only if for all cfgs G, t(L(G))=L(G).

Theorem 4.6: Let T be any set of grammatical transformations on the cfgs with terminal alphabets equal to $\hat{\Sigma}$ such that
(i) for all t in T, t is language preserving; and
(ii) the identity transformation is in T.
Let Γ be any class of cfgs such that
(iii) $\mathscr{C} \subseteq \Gamma$; and
(iv) Γ contains no inherently ambiguous cfgs.
Then for an arbitrary cfg G, it is undecidable if there exists t in T such that t(G) is an element of Γ.

Proof: The proof of Theorem 4.6 follows easily from Theorem 4.2 and is left to the reader.

Finally we note that similar results hold for many classes of transformations that do not contain the identity transformation, e.g. the transformations in [16].

5. Conclusions

A complexity theory for many different grammar problems was presented. Sufficient conditions on a class of languages F, for the undecidability of the F and F H-separability problems, were presented. Most classes of languages studied in the literature satisfy these conditions. Sufficient conditions on grammar classes Γ and relations ρ, for the undecidability of the predicate "For an arbitrary cfg G, there exists a cfg H in Γ for which GρH.", were presented. Relations satisfying these conditions include equivalence, structural equivalence, and covering; grammar classes satisfying these conditions include the classes of strong LL, LL, strong LC, LC, SLR, LALR, LR, and BRC grammars.

Bibliography

[1] A.V. Aho and J.D. Ullman, The Theory of Parsing, Translation, and Compiling, Vols. 1 and 2, Prentice-Hall, Englewood Cliffs, N.J., 1972 and 1973.

[2] K. Culik, II and R. Cohen, LR-regular grammars -- an extension of LR(k) grammars, JCSS 7, 1(1973), 66-96.

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

[4] H.B. Hunt, III and T.G. Szymanski, Complexity metatheorems for context-free grammar problems, Princeton University Computer Science Technical Report 187, July 1975 (also submitted for publication).

[5] J. Hartmanis and J.E. Hopcroft, An overview of the theory of computational complexity, JACM 18, 13(1971), 444-475.

[6] K. Culik, II and R. Cohen, LR-regular grammars -- an extension of LR(k) grammars, JCSS 7, 1 (1973), 66-96.

[7] T.G. Szymanski, Generalized bottom-up parsing, Ph.D. thesis, Cornell University, Ithaca, New York, May 1973.

[8] W.F. Ogden, unpublished note, December 1974.

[9] R. McNaughton and S. Papert, Counter-Free Automata, MIT Press, Cambridge, Massachusetts, 1971.

[10] A.J. Korenjak and J.E. Hopcroft, Simple deterministic languages, IEEE Conference Record of 7th Annual Symposium on Switching and Automata Theory, October 1966, 36-46.

[11] M.A. Harrison and I.M. Havel, Strict deterministic grammars, JCSS 7, 3(1973), 237-277.

[12] H.B. Hunt, III and J.L. Rangel, Decidability of equivalence, containment, intersection, and separability of context-free languages, Center for Research in Computing Technology TR 19-75, Harvard University, Cambridge, Massachusetts, October 1975.

[13] D.E. Knuth, A characterization of parenthesis languages, Inf. and Control, 11(1967), 269-289.

[14] H.B. Hunt, III, T.G. Szymanski, and J.D. Ullman, Operations on sparse relations and efficient algorithms for grammar problems, Proc. 15th Annual IEEE Symp. on Switching and Automata Theory, October 1974, 127-132.

[15] ________, On the complexity of LR(k) testing, Conference Record 2nd ACM Symp. on Principles of Programming Languages, January 1975, 130-136.

[16] B.M. Brosgol, Deterministic translation grammars, Ph.D. thesis, Harvard University, Cambridge, Massachusetts, 1974.

[17] C.N. Fischer, On parsing context-free languages in parallel environments, Ph.D. thesis, Cornell University, Ithaca, New York, April 1975.

[18] T.G. Szymanski and J.H. Williams, Non-canonical extensions of bottom-up parsing techniques, to appear in SIAM Journal on Computing.