On the Decidability of Grammar Problems

H. B. HUNT, III

State University of New York at Albany, Albany, New York

ABSTRACT The decidability of context-free grammar and language problems in general, especially problems about separability and context, grammar and language class membership, and grammatical similarity relations, is studied. The results are based upon efficient reductions of membership problems for always-halting Turing machines and the following fundamental property of s-grammars and s-languages:

The emptiness-of-intersection problem for pairs of s-grammars that generate languages without arbitrarily long common prefixes is decidable but is *not* recursively time-bounded.

Categories and Subject Descriptors. F.1.1 [**Computation by Abstract Devices**]: Models of Computation—*computability theory*; F.4.2 [**Mathematical Logic and Formal Languages**]. Grammars and Other Rewriting Systems—*decision problems; grammar types*; F.4.3 [**Mathematical Logic and Formal Languages**]: Formal Languages—*classes defined by grammars or automata; decision problems*

General Terms: Languages, Theory

Additional Key Words and Phrases: Context-free grammar, context-free language, decidability, separability, context, similarity relation, grammatical covering

1. *Introduction*

The close interrelationship among programming language syntax, context-free grammars and languages, parsing, and compiling is well known and is extensively discussed in [1]. Unfortunately, many of the problems concerning programming languages, parsing, or compiling that one might wish to solve are equivalent to undecidable or computationally intractable context-free grammar or language problems. Several such problems are

(1) the *emptiness-of-intersection problem*, that is, the problem of determining if the intersection of the languages generated by two grammars is empty;

(2) *grammar-class-membership problems*, that is, problems of determining, for a grammar G and grammar class Γ, whether G is an element of Γ;

(3) *language-class-membership problems*, that is, problems of determining, for a grammar G and class of languages $\mathscr{L}$, whether the language generated by G is an element of $\mathscr{L}$, and

(4) *grammatical-similarity-relation problems*, that is, problems of determining, for a grammar G, binary relation ρ, and grammar class Γ, whether there exists a grammar $H \in \Gamma$ such that $G \rho H$.

This research was supported in part by the National Science Foundation under Grants DCR 75-22505 and MCS 77-27197.

A preliminary version of some of these results was presented at the 3rd ACM Symposium on Principles of Programming Languages, Atlanta, Ga., Jan 1976.

Author's address: Department of Computer Science, State University of New York at Albany, 1400 Washington Avenue, Albany, NY 12222.

Here we study the decidability of context-free grammar and language problems in general, especially problems related to (1)–(4).

Most previous work on the undecidability of context-free grammar and language problems has involved effective reductions of variants of the Post Correspondence Problem or the halting problem for Turing machines. In [11] and [12], however, we have shown that efficient, not just effective, reductions of membership problems for *always-halting* Turing machines can also be used to derive undecidability results. Frequently it is technically easier to embed always-halting Turing machine computations efficiently than it is to embed arbitrary Turing machine computations effectively. In Section 2 we use efficient reductions of membership problems for always-halting Turing machines to prove three new facts about the complexity of s-grammars and s-languages. In subsequent sections these three facts are shown to imply much, if not most, of what was previously known about the undecidability of context-free grammar and language problems. Furthermore, they are used to derive many new results as well.

Separability was introduced in [2] in an attempt to partially circumvent the undecidability of grammar problems such as the emptiness-of-intersection problem. Intuitively, the idea of separability is to replace arbitrary context-free languages by approximating languages for which such problems are decidable. In Section 3, using the s-grammar complexity results of Section 2, we investigate the inherent limitations of separability. We show that separability is undecidable in general for the *same* reason that the emptiness-of-intersection problem is undecidable. Therefore, it is unlikely that separability can be used to circumvent the undecidability of the emptiness-of-intersection problem. Letting Γ denote *any* class of grammars containing all s-grammars, a general class $\mathscr{C}[\Gamma]$ of separability problems for pairs of grammars in Γ is introduced, and each of its members is shown to be undecidable. The class $\mathscr{C}[\Gamma]$ includes the regular separability problem introduced in [2]; the problems of determining whether the intersection of the languages generated by two grammars is empty, finite, regular, or context-free; and the problem of determining whether the languages generated by two grammars have arbitrarily long common prefixes.

In Section 4 we study the decidability of grammatical-similarity-relation problems. The s-grammar complexity results of Section 2 are shown to imply directly and easily the undecidability of a variety of such problems. A single theorem is presented that yields sufficient conditions for the undecidability of grammatical-similarity problems for a variety of relations and grammar classes studied in the literature. These relations include equivalence, language equivalence, structural equivalence, structural containment, grammatical covering, Reynolds covering, and grammatical homomorphisms. These grammar classes include the LL, strong LL, LR, SLR, LALR, LL-regular [14], LR-regular [2], FSPA [21], and unambiguous grammars.

In Section 5 we show that the s-grammar complexity results of Section 2 also imply the undecidability of a variety of grammatical covering problems.

Section 6 is a short conclusion.

The remainder of this section consists of standard definitions and basic results about binary relations, context-free grammars and languages, pushdown automata, and Turing machine computations needed to read this paper. We assume that the reader is familiar with the basic facts and results about context-free grammars and languages, Turing machines, and decidability as presented in [1] or [9]. Henceforth we abbreviate context-free grammar by cfg, context-free language by cfl, deterministic pushdown automata by dpda, and Turing machine by Tm. We denote the empty string by λ, the empty set by $\varnothing$, and the length of a string T or the cardinality of a set T by $|T|$. We denote the set of natural numbers by $\mathscr{N}$.

Definition 1.1. Let T be a nonempty set. Let ρ, σ, and τ be binary relations on T such that for all $x, y \in T$, if $x \ \rho \ y$, then $x \ \sigma \ y$, and if $x \ \sigma \ y$, then $x \ \tau \ y$. Then the relation σ is said to be *between the relations* ρ and τ.

Definition 1.2. A *context-free grammar* is a 4-tuple $G = (V, \Sigma, P, S)$, where V and Σ are disjoint nonempty sets of *nonterminals* and *terminals*, respectively, $S \in V$ is the *start symbol*, and $P \subsetneq V \times (V \cup \Sigma)^*$ is a finite set of *productions*. We write $A \to w$ in P instead of $(A, w) \in P$. The *size of* G, denoted by $|G|$, equals the sum of $|A \cdot w|$ taken over all productions $A \to w$ of G.

A cfg $G = (V, \Sigma, P, S)$ is said to be an *s-grammar* if and only if for all productions $A \to w$ in P, $w \in \Sigma\cdot(V \cup \Sigma)^*$, and for all $A \in V$ and $a \in \Sigma$, there is at most one production of the form $A \to a\cdot\omega$ in P, where $\omega \in (V \cup \Sigma)^*$. We denote the class of all s-grammars with terminal alphabet $\{0, 1\}$ by $\mathscr{S}$.

Definition 1.3. Let $G = (V, \Sigma, P, S)$ be a cfg. We define the binary relation $\Rightarrow_G$ on $(V \cup \Sigma)^*$ by $\phi \Rightarrow_G \psi$, for all $\phi, \psi \in (V \cup \Sigma)^*$ (read ϕ *derives* ψ *in* G) if and only if there exist $\alpha, \beta \in (V \cup \Sigma)^*$ and $A \to w$ in P such that $\phi = \alpha\cdot A\cdot\beta$ and $\psi = \alpha\cdot w\cdot\beta$. We denote the transitive reflexive closure of $\Rightarrow_G$ by $\Rightarrow_G^*$. The subscript G is omitted whenever the identity of the grammar G is clear. If $\alpha(\beta)$ consists solely of terminals, we say that ϕ *derives* ψ in G *by a leftmost* (*rightmost*) *derivation*, denoted by $\phi \Rightarrow_{G,\text{lm}} \psi$ ($\phi \Rightarrow_{G,\text{rm}} \psi$). If ϕ derives ψ in G by a leftmost derivation using a specific sequence π of productions, we denote this by $\phi \Rightarrow_{G,\text{lm}}^{\pi} \psi$.

The language of G, denoted by $L(G)$, equals $\{x \in \Sigma^* \mid S \Rightarrow_G^* x\}$. Two cfgs G and H are said to be *equivalent* if and only if $L(G) = L(H)$.

In the next several definitions and propositions, $G = (N, \Sigma, P, S)$ is a cfg. Henceforth, whenever necessary we assume that all cfgs are reduced.

Definition 1.4. Let k be a nonnegative integer. Let $\alpha \in (N \cup \Sigma)^*$. We define the sets $\text{First}_k(\alpha)$, $\text{Eff}_k(\alpha)$, and $\text{Follow}_k(\alpha)$ by

(1) $\text{First}_k(\alpha) = \{x \in \Sigma^* \mid \alpha \Rightarrow^* x\cdot\beta \text{ and } |x| = k, \text{ or } \alpha \Rightarrow^* x \text{ and } |x| < k\}$;
(2) $\text{Eff}_k(\alpha) = \{x \in \Sigma^* \mid x \in \text{First}_k(\alpha)$ and there exists a derivation $\alpha \Rightarrow_{\text{rm}}^* \beta \Rightarrow_{\text{rm}} x\cdot y$ in which $\beta \neq A\cdot x\cdot y$ for any nonterminal $A\}$; and
(3) $\text{Follow}_k(\alpha) = \{x \in \Sigma^* \mid S \Rightarrow^* w\cdot\alpha\cdot x\cdot y \text{ with } |x| = k, \text{ or } S \Rightarrow^* w\cdot\alpha\cdot x \text{ with } |x| < k\}$.

The above definitions can be extended in the usual way to apply to sets of strings. The next three definitions are taken from [1].

Definition 1.5. Let k be a positive integer. A cfg G is said to be a *strong* $LL(k)$ *grammar* if and only if, for all $A \in N$, if $A \to \beta$ and $A \to \gamma$ are distinct productions of G, then $\text{First}_k(\beta\cdot\text{Follow}_k(A)) \cap \text{First}_k(\gamma\cdot\text{Follow}_k(A)) = \varnothing$.

Definition 1.6. Let k be a nonnegative integer. If $S \Rightarrow_{\text{rm}}^* \beta\cdot A\cdot w \Rightarrow_{\text{rm}} \beta\cdot\alpha\cdot w$, then the prefix $\beta\cdot\alpha$ is said to be a *viable prefix of* G. We say that $[A \to \beta_1.\beta_2, u]$ is an $LR(k)$ *item of* G if $A \to \beta_1\cdot\beta_2 \in P$ and $u \in \text{First}_k(\Sigma^*)$. We say that $[A \to \beta_1.\beta_2, u]$ is *valid for* $\alpha\cdot\beta_1$, a viable prefix of G, if there is a derivation

$$S \Rightarrow_{\text{rm}}^* \alpha\cdot A\cdot w \Rightarrow_{\text{rm}} \alpha\cdot\beta_1\cdot\beta_2\cdot w,$$

for which $u \in \text{First}_k(w)$. A cfg G is said to be an $SLR(k)$ *grammar* if and only if for every viable prefix γ of G, whenever $[A \to \alpha., \lambda]$ and $[B \to \beta_1.\beta_2, \lambda]$ are both valid LR(0) items for γ, then $\text{Follow}_k(A) \cap \text{Eff}_k(\beta_2\cdot\text{Follow}_k(B)) = \varnothing$.

Definition 1.7. Let m and n be nonnegative integers. A cfg G is said to be an *(m, n)-bounded right-context* (*BRC*) *grammar* if and only if the four conditions,

(1) $\$^m \cdot S' \cdot \$^n \Rightarrow^*_{G',\mathrm{rm}} \alpha \cdot A \cdot w \Rightarrow_{G',\mathrm{rm}} \alpha \cdot \beta \cdot w$,
(2) $\$^m \cdot S' \cdot \$^n \Rightarrow^*_{G',\mathrm{rm}} \gamma \cdot B \cdot w \Rightarrow_{G',\mathrm{rm}} \gamma \cdot \delta \cdot x = \alpha' \cdot \beta \cdot y$ are rightmost derivations in the augmented grammar $G' = (N \cup \{S'\}, \Sigma, P \cup \{S' \to S\}, S')$,
(3) $|x| \leq |y|$, and
(4) the last m symbols of α and α' coincide, and the first n symbols of w and y coincide,

imply that $\alpha' = \gamma$, $A = B$, and $y = x$

The definition of the class of strict deterministic grammars can be found in [8].

Definition 1.8. [1, 22]. A cfg G is said to be *ambiguous* if and only if some string in $L(G)$ has two distinct leftmost derivations. A cfg G is said to be *inherently ambiguous* if and only if every equivalent cfg is ambiguous. A cfg G is said to be *structurally ambiguous* if and only if some string in $L(G)$ has two distinct derivation trees that are the same except for nonterminal labels.

We consider several grammatical similarity relations studied in the literature placing special emphasis on grammatical covering.

Definition 1.9 [1, 4]. Let $G = (N, \Sigma, P, S)$ and $H = (M, \Sigma, Q, T)$ be cfgs. We say that *H left covers G* if there exists a homomorphism h from Q^* to P^* such that

(a) if $T \Rightarrow^{\pi}_{H,\mathrm{lm}} w$ where $w \in \Sigma^*$, then $S \Rightarrow^{h(\pi)}_{G,\mathrm{lm}} w$, and
(b) for all derivations π such that $S \Rightarrow^{\pi}_{G,\mathrm{lm}} w$ where $w \in \Sigma^*$, there exists a derivation π' such that $T \Rightarrow^{\pi'}_{H,\mathrm{lm}} w$ and $h(\pi') = \pi$.

If, in addition, the homomorphism h is λ-free, that is, for all $a \in Q$, $h(a) \neq \lambda$, we say that *H λ-free left covers G*. *Right covering* and *λ-free right covering* are defined analogously.

If H left or right covers G (or if H λ-free left or λ-free right covers G), we say that *H covers G* (or *H λ-free covers G*).

The other grammatical-similarity relations considered here include structurally equivalence and containment defined in [1, 18], grammatical homomorphisms and isomorphisms defined in [4], and Reynolds covering defined in [4, 19].

Definition 1.10. A language L is said to be *prefix-free* if for all strings $x, y \in L$ such that $x = y \cdot z$, either $y = \lambda$ or $z = \lambda$.

Definition 1.11. A language L is said to be a *definite event* if and only if there exists a finite alphabet Σ and finite sets $F_1 \subseteq \Sigma^*$ and $F_2 \subseteq \Sigma^*$ such that $L = F_1 \cdot \Sigma^* \cup F_2$.

Definition 1.12. Let $\mathscr{F}$ be a class of languages. We say that two languages L_1 and L_2 are

(i) *$\mathscr{F}$ separable* if and only if there exists a language $S \in \mathscr{F}$ such that $L_1 \subseteq S$ and $L_2 \cap S = \varnothing$ or $L_1 \cap S = \varnothing$ and $L_2 \subseteq S$, and are
(ii) *$\mathscr{F}H$-separable* if and only if there exist disjoint languages S_1 and S_2 in $\mathscr{F}$ such that $L_1 \subseteq S_1$ and $L_2 \subseteq S_2$.

Let Γ be a class of grammars. The *regular set* and *definite event separability problems for* Γ are the problems of determining, for grammars G and H in Γ, whether the languages $L(G)$ and $L(H)$ are $\mathscr{F}$ separable, where $\mathscr{F}$ is the class of all regular sets and all definite events, respectively.

Definition 1.13. Let M be a deterministic single tape Tm with tape alphabet T, state set S, and set of accepting states F. An *instantaneous description* (i.d.) of M is any string in $T^*\cdot(S \times T)\cdot T^*$.

We define the binary relation $\vdash_M$ on the set of all i.d.s of M as in [9]. Thus $\alpha \vdash_M \beta$ if and only if i.d. β follows from i.d. α by one application of the state transition function of M.

2. *Basic S-Grammar Complexity Results*

In this section we derive the basic s-grammar and s-language complexity results used in this paper.

Let M be a single-tape deterministic Tm with state set Q, tape alphabet T, start state q_0, and final state q_f that

(1) always adds new tape squares on the right end of its tape, and
(2) always halts on the right end of its tape.

Given an input string $x = x_1\cdot x_2\cdot \cdots \cdot x_n$ to M, two s-grammars $G_1[M, x]$ and $G_2[M, x]$ (presented in Appendix A) can be constructed in linear time on a deterministic multitape Tm such that

(a) $L(G_1[M, x]) = \{\#\cdot(q_0, x_1)\cdot x_2\cdot \cdots \cdot x_{n-1}\cdot \hat{x}_n\cdot\#\}\cdot[\{\hat{b}\cdot y^{\text{rev}}\cdot\#\cdot y\cdot\hat{b} \mid y\cdot b$ is an i.d. of M that is not an accepting i.d. of $M\}\cdot\{\#\}]^*\cdot\{(q_f, \hat{a})\cdot z\cdot\# \mid z^{\text{rev}}\cdot(q_f, a)$ is an accepting i.d. of $M\}\cdot\{\$\}$, and
(b) $L(G_2[M, x]) = \{\#\}\cdot[\{y\cdot\hat{a}\cdot\#\cdot\hat{a}\cdot z^{\text{rev}} \mid y$ and z are i.d.'s of M of length ≥ 3 and $y \vdash_M z\}\cdot\{\#\}]^*\cdot\{\$\}$.

Clearly, $L(G_1[M, x]) \cap L(G_2[M, x]) \neq \varnothing$ if and only if $x \in L(M)$. Moreover, if M halts for input x, then the languages $L(G_1[M, x])$ and $L(G_2[M, x])$ do *not* have arbitrarily long common prefixes. We use these simple observations to prove three basic facts about s-grammars that imply the undecidability and/or nonrecursive time complexity of a variety of problems concerning context-free grammars and languages. To prove these facts, we need the following two well-known properties of recursive sets and Tms that always halt.

PROPOSITION 2.1. *For all recursive functions f, there exists a recursive set S_f such that any Tm that recognizes S_f makes more than $f(n)$ moves before halting, for infinitely many n.*

PROPOSITION 2.2. *For every deterministic Tm M that halts for all inputs there exists a strictly increasing recursive function $T(n)$ such that for all inputs of size n, M makes no more than $T(n)$ moves before halting.*

THEOREM 2.3

(1) *The emptiness-of-intersection problem for pairs of s-grammars that generate languages without arbitrarily long common prefixes is decidable but is* not *recursively time-bounded.*
(2) *For all recursive functions $f: \mathcal{N} \to \mathcal{N}$ there exists a pair (G, H) of s-grammars that generate languages without arbitrarily long common prefixes, such that $L(G) \cap L(H) \neq \varnothing$ but $f(|G| + |H|) < |x|$ for all strings x in $L(G) \cap L(H)$.*

(3) *For all pairs* (G, H) *of s-grammars that generate languages without arbitrarily long common prefixes, the language* $L(G)\cdot\{\dashv\} \cup L(H)\cdot\{\dashv\}$ *is also an s-language, where* $\dashv$ *is any symbol not appearing in a word in* $L(G)$ *or* $L(H)$.

PROOF

(1) The proof that the problem is *not* recursively time-bounded is by contradiction. Suppose that the emptiness-of-intersection problem for pairs of s-grammars that generate languages without arbitrarily long common prefixes is recursively time-bounded. Then by Proposition 2.2 there is a strictly increasing recursive function $f:\mathcal{N}\to\mathcal{N}$ that bounds above the time required to decide it on some deterministic Tm. Let M be any always-halting single-tape deterministic Tm satisfying conditions (1) and (2) above. We claim that the language $L(M)$ is recognizable by a $c\cdot n + f(c\cdot n)$ time-bounded deterministic Tm, where c is a constant depending only upon M.

Let $\mathcal{M}$ be the deterministic Tm that operates as follows.

(a) Given input x to M, the Tm $\mathcal{M}$ constructs the s-grammars $G_1[M, x]$ and $G_2[M, x]$ described above.
(b) $\mathcal{M}$ tests if $L(G_1[M, x]) \cap L(G_2[M, x]) \neq \emptyset$. If so, $\mathcal{M}$ accepts x. Otherwise, $\mathcal{M}$ rejects x.

Step (1) requires at most $O(|x|)$ time. Since M always halts by assumption for all inputs x to M, the s-languages $L(G_1[M, x])$ and $L(G_2[M, x])$ do not have arbitrarily long common prefixes. Thus, by assumption, step (2) requires at most $f(|G_1[M, x]| + |G_2[M, x]|)$ time. But $f(|G_1[M, x]| + |G_2[M, x]|) < f(c\cdot|x|)$ for all sufficiently large constants c. Thus, as claimed, the language $L(M)$ is recognizable by a $c\cdot n + f(c\cdot n)$ time-bounded deterministic Tm.

Finally, for all positive integers a the recursive function $F(n) = n^2 + f(n^2)$ is strictly greater than the function $a\cdot n + f(a\cdot n)$, for all but finitely many positive integers n. But this implies that every recursive set is recognizable by a Tm that makes no more than $F(n)$ moves for all but finitely many n, contradicting Proposition 2.1.

The proof that the emptiness-of-intersection problem for pairs of s-grammars that generate languages without arbitrarily long common prefixes is decidable is easy and is left to the reader.

(2) If not, the emptiness-of-intersection problem for pairs of s-grammars that generate languages without arbitrarily long common prefixes would be recursively time-bounded, contradicting assertion (1).

(3) Let $G = (V_1, \Sigma_1, P_1, S_1)$ and $H = (V_2, \Sigma_2, P_2, S_2)$ be s-grammars, and generate languages without arbitrarily long common prefixes. Without loss of generality we assume that both grammars G and H are reduced, $V_1 \cap V_2 = \emptyset$, and exactly one terminal symbol appears in the right-hand side of any production in P_1 or in P_2. The idea is to construct an s-grammar that simultaneously simulates the leftmost derivations of G and H until the prefix generated uniquely determines which grammar must generate the string or until the end marker $\dashv$ is generated. This is possible because the languages $L(G)$ and $L(H)$ do *not* have arbitrarily long common prefixes. □

Several words on the significance of Theorem 2.3 are in order. The technique of constructing two context-free grammars G and H such that the language $L(G) \cap L(H)$ denotes a Turing-machine computation is well known [6]. The importance and novelty of Theorem 2.3 lies in the two facts that

(1) the grammars $G_1[M, x]$ and $G_2[M, x]$ are s-grammars, and
(2) the grammars $G_1[M, x]$ and $G_2[M, x]$ generate languages without arbitrarily long common prefixes.

These two properties of $G_1[M, x]$ and $G_2[M, x]$ hold because we restricted attention to terminating Turing-machine computations. The importance of these two properties, especially the second, will become clear in the rest of this paper.

We conclude Section 2 by noting that

(i) standard encoding techniques can be used to show that Theorem 2.3(1) holds even if the s-grammars are known to be elements of $\mathcal{S}$, and

(ii) a slight modification of the construction in the proof of Theorem 2.3(3) can be used to show the following: For all pairs (G, H) of s-grammars that generate languages without arbitrarily long common prefixes, the languages $L(G)\cdot\{a\} \cup L(H)\cdot\{b\}$ is also an s-language, where a and b are distinct symbols not appearing in any string in $L(G)$ or $L(H)$.

3. *Separability, Context, and Emptiness of Intersection*

Several authors [2, 14, 21] have attempted to extend the concepts of LL or LR parsing by developing techniques that rely upon some sort of "separating" set to determine the local context of a string being parsed. Culik and Cohen [2] showed that the membership problem for the class of LR-regular grammars is effectively reducible to the regular separability problem for the cfgs. Szymanski [20] showed that the regular separability problem for the LR(1) grammars is effectively reducible to the membership problem for the class of FSPA(1) grammars. Ogden [17] showed that the membership problem for the class of LR-regular grammars is undecidable and thus that the regular separability problems for the cfgs is undecidable. Here, letting Γ denote *any* class of grammars containing $\mathcal{S}$, we present a general class $\mathscr{C}[\Gamma]$ of separability problems for pairs of grammars in Γ and show that each problem in $\mathscr{C}[\Gamma]$ is undecidable for the same reason.

THEOREM 3.1. *Let ρ_1 and ρ_2 be the binary relations on $\mathcal{S}$ defined, for all grammars G and H in $\mathcal{S}$, by*

(i) $G \rho_1 H$ if and only if the languages $L(G)$ and $L(H)$ are definite event separable; and
(ii) $G \rho_2 H$ if and only if the intersection of the languages $L(G)$ and $L(H)$ is a context-free language.

Let ρ be any binary relation on $\mathcal{S}$ between ρ_1 and ρ_2. Then it is undecidable, for grammars G and H in $\mathcal{S}$, whether $G \rho H$.

PROOF. Let ρ be any binary relation on $\mathcal{S}$ between the binary relations ρ_1 and ρ_2. We show that there is *no* recursive function $f: \mathcal{N} \to \mathcal{N}$ such that, letting G and H be arbitrary grammars in $\mathcal{S}$ that generate languages without arbitrarily long common prefixes, $f(|G| + |H|)$ bounds above the number of moves required by some deterministic Tm to determine if $G \rho H$. Since the set $\mathcal{S}$ is recursive, this implies that the problem of determining whether $G \rho H$, for grammars G and H in $\mathcal{S}$, is undecidable. The proof is by contradiction.

Suppose such a recursive function f exists. Without loss of generality we assume that f is increasing. Let $G = (N_1, \{0, 1\}, P_1, S_1)$ and $H = (N_2, \{0, 1\}, P_2, S_2)$ be arbitrary grammars in $\mathcal{S}$ that generate languages without arbitrarily long common prefixes. Let S, B_1, B_2, C_1, C_2, $\$$, $\cent$, a, b, and c be distinct symbols that are not elements of $N_1 \cup N_2 \cup \{0, 1\}$. Let

$$G' = (N_1 \cup \{S, B_1, B_2\}, \{0, 1, \$, \cent, a, b, c\}, P_1', S),$$

where

$$P_1' = P_1 \cup \{S \to \$\cdot S_1\cdot \cent\cdot a\cdot B_1\cdot b\cdot \cent\cdot c\cdot B_2,\\ B_1 \to a\cdot B_1\cdot b,\ B_1 \to \cent,\ B_2 \to c\cdot B_2,\ B_2 \to \cent\}.$$

Let

$$H' = (N_2 \cup \{S, C_1, C_2\}, \{0, 1, \$, \cent, a, b, c\}, P_2', S),$$

where

$$P_2' = P_2 \cup \{S \to \$\cdot S_2\cdot \cent\cdot a\cdot C_1\cdot b\cdot C_2\cdot c\cdot \cent,\\ C_1 \to a\cdot C_1,\ C_1 \to \cent,\ C_2 \to b\cdot C_2\cdot c,\ C_2 \to \cent\}.$$

Clearly, the cfgs G' and H' are s-grammars. Also, clearly,

$$L(G') = \{\$\}\cdot L(G)\cdot \{\cent\cdot a^n\cdot \cent\cdot b^n\cdot \cent\cdot c^m\cdot \cent \mid n, m \geq 1\},$$

$$L(H') = \{\$\}\cdot L(H)\cdot \{\cent\cdot a^n\cdot \cent\cdot b^m\cdot \cent\cdot c^m\cdot \cent \mid n, m \geq 1\}.$$

We claim that

(i) if $L(G) \cap L(H) = \varnothing$, then the languages $L(G')$ and $L(H')$ are definite event separable, and
(ii) if $L(G) \cap L(H) \neq \varnothing$, then the language $L(G') \cap L(H')$ is not a context-free language.

Claim (i) follows, since by assumption the languages $L(G)$ and $L(H)$ do not have arbitrarily long common prefixes. Claim (ii) follows by noting that if there exists a string $w \in L(G) \cap L(H)$, then the language $[L(G') \cap L(H')] \cap \{\$\cdot w\cdot \cent\}\cdot \{a, b, c, \cent\}^* = \{\$\cdot w\cdot \cent\cdot a^n\cdot \cent\cdot b^n\cdot \cent\cdot c^n\cdot \cent \mid n \geq 1\}$, which is not a context-free language. Since the context-free languages are closed under intersection with regular sets, this implies that the language $L(G') \cap L(H')$ is also not a context-free language.

Finally, using standard encoding arguments, we can encode the s-grammars G' and H' into grammars G'' and H'', respectively, in $\mathscr{S}$, such that claims (i) and (ii) still hold when G'' and H'' are substituted for G' and H', respectively. Clearly, there exists an increasing recursive function $g: \mathscr{N} \to \mathscr{N}$ such that G'' is constructible from G and H'' is constructible from H within times $g(|G|)$ and $g(|H|)$, respectively, on some deterministic Tm. But this implies that the emptiness-of-intersection problem for pairs of grammars in $\mathscr{S}$ that generate languages without arbitrarily long common prefixes is recursively time-bounded, contradicting Theorem 2.3(1). □

In the next three corollaries we present examples of the separability problems that satisfy the conditions of Theorem 3.1.

COROLLARY 3.2. *Let $\mathscr{F}$ be any class of languages containing all definite events over $\{0, 1\}$. Then it is undecidable, for grammars G and H in $\mathscr{S}$,*

(1) *whether the languages $L(G)$ and $L(H)$ are $\mathscr{F}$-separable, and*
(2) *whether the languages $L(G)$ and $L(H)$ are $\mathscr{F}H$-separable.*

PROOF. Let $\mathscr{F}$ be any class of languages containing all definite events over $\{0, 1\}$. Let ρ and σ be the binary relations on $\mathscr{S}$ defined by, for all grammars G and

H in $\mathscr{S}$,

(i) $G \rho H$ if and only if the languages $L(G)$ and $L(H)$ are $\mathscr{F}$ separable, and
(ii) $G \sigma H$ if and only if the languages $L(G)$ and $L(H)$ are $\mathscr{F}H$-separable.

Let ρ_1 and ρ_2 be the binary relations on $\mathscr{S}$ defined in the statement of Theorem 3.1. Since $\mathscr{F}$ contains all definite events over $\{0, 1\}$ by assumption and the definite events over $\{0, 1\}$ are closed under complementation, if $G \rho_1 H$, then $G \rho H$ and $G \sigma H$. Moreover, if $G \rho H$ or if $G \sigma H$, then $L(G) \cap L(H) = \varnothing$ and $G \rho_2 H$. Thus the relations ρ and σ are between the relations ρ_1 and ρ_2. Hence, by Theorem 3.1 it is undecidable for grammars G and H in $\mathscr{S}$ whether the languages $L(G)$ and $L(H)$ are $\mathscr{F}$ separable, or the languages $L(G)$ and $L(H)$ are $\mathscr{F}H$-separable. □

COROLLARY 3.3. *It is undecidable for grammars G and H in $\mathscr{S}$ whether the languages $L(G)$ and $L(H)$ have arbitrarily long common prefixes.*

PROOF. Let ρ be the binary relation on $\mathscr{S}$ defined, for all grammars G and H in $\mathscr{S}$, by $G \rho H$ if and only if the languages $L(G)$ and $L(H)$ do not have arbitrarily long common prefixes. For all grammars G and H in $\mathscr{S}$,

(1) if the languages $L(G)$ and $L(H)$ are definite event separable, then they do not have arbitrarily long common prefixes, and
(2) if the languages $L(G)$ and $L(H)$ do not have arbitrarily long common prefixes, then the language $L(G) \cap L(H)$ is finite and hence is a context-free language.

Thus the relation ρ is between the relations ρ_1 and ρ_2 defined in the statement of Theorem 3.1. Hence, by Theorem 3.1 it is undecidable, for grammars G and H in $\mathscr{S}$, whether the languages $L(G)$ and $L(H)$ have arbitrarily long common prefixes. □

COROLLARY 3.4. *Let $\mathscr{F}$ be any subset of the cfls over $\{0, 1\}$ containing the empty set. Then it is undecidable for grammars G and H in $\mathscr{S}$ whether the language $L(G) \cap L(H)$ is an element of $\mathscr{F}$. In particular, it is undecidable whether the language $L(G) \cap L(H)$ is empty, finite, regular, or context-free.*

PROOF. Let $\mathscr{F}$ be any subset of the cfls over $\{0, 1\}$ containing the empty set. Let ρ be the binary relation on $\mathscr{S}$ defined, for all grammars G and H in $\mathscr{S}$, by $G \rho H$ if and only if $L(G) \cap L(H) \in \mathscr{F}$. Then the relation ρ is between the relations ρ_1 and ρ_2 defined in the statement of Theorem 3.1. Hence, by Theorem 3.1 it is undecidable for grammars G and H in $\mathscr{S}$ whether $G \rho H$ or, equivalently, whether $L(G) \cap L(H) \in \mathscr{F}$. □

Theorem 3.1 and its corollaries show that *any* attempt to generalize the LL or the LR parsing techniques through the use of more complex right context must yield a grammar class with an undecidable membership problem. Moreover, they can be used to prove the undecidability of other grammar problems such as

(1) determining if a cfg is FSPA(1) [21];
(2) determining if an LR(1) grammar is SLR [13]; and
(3) determining if an LL(2) grammar is strong LL [13].

4. *Grammatical Similarity Problems*

In this section we show how the s-grammar complexity results of Section 2 can be efficiently embedded into grammatical-similarity-relation problems. To accomplish this, we introduce the concepts of tree and inherent tree ambiguity.

Definition 4.1. Let $G = (N, \Sigma, P, S)$ be a cfg. We denote the cfg $(N \cup N', \Sigma, P', S)$, where

(1) $N' = \{\overline{[A]} \mid A \in N\}$ is a set of new nonterminals, and
(2) $P' = \{A \to \overline{[A]}\cdot\alpha \mid A \to \alpha \text{ is in } P\} \cup \{A \to \lambda \mid A \in N'\}$, by $T_1(G)$.

As shown in [5], if G is strict deterministic, then the cfg $T_1(G)$ is simultaneously strict deterministic and BRC.

We present a construction, simple variants of which suffice to directly embed the emptiness-of-intersection problem for s-grammars that generate languages without arbitrarily long common prefixes into a variety of grammar class or language class membership problems. Let $G_1 = (M_0, \Sigma, P_0, S_0)$ and $H_1 = (N_0, \Sigma, Q_0, T_0)$ be fixed cfgs. Let $G = (M', \{0, 1\}, P', S')$ and $H = (N', \{0, 1\}, Q', T')$ be arbitrary grammars in $\mathcal{S}$ such that $M_0 \cap M' = \varnothing$ and $N_0 \cap N' = \varnothing$. Let A, \$', \$'', \$, and ¢ be distinct symbols not appearing in G, H, G_1, or H_1, where $\Delta = \Sigma \cup \{0, 1, \$, ¢\}$ and A, \$', and \$'' are nonterminal symbols.

We define the cfgs $\hat{G}[G, G_1]$, $\hat{H}[H, H_1]$, $\hat{G}'[G, H]$, $\hat{H}'[G, H]$, and $\mathscr{G}[G, H, G_1, H_1]$ as follows:

(1) $\hat{G}[G, G_1] = (\hat{M}, \Delta, \hat{P}, A)$, where

  (i) $\hat{M} = M' \cup M_0 \cup \{A, \$'\}$, and
  (ii) $\hat{P} = P' \cup P_0 \cup \{A \to \$'\cdot\$\cdot S'\cdot ¢\cdot S_0 ¢, \$' \to \lambda\}$.

(2) $\hat{H}[H, H_1] = (\hat{N}, \Delta, \hat{Q}, A)$, where

  (i) $\hat{N} = N' \cup N_0 \cup \{A\} \cup \{A, \$''\}$, and
  (ii) $\hat{Q} = Q' \cup Q_0 \cup \{A \to \$''\cdot\$\cdot T'\cdot ¢\cdot T_0\cdot ¢, \$'' \to \lambda\}$.

We assume that $\hat{M} \cap \hat{N} = \{A\}$.

(3) $\hat{G}'[G, H] = T_1(\hat{G}[G, G_1]) = (M, \Delta, P, A)$.
(4) $\hat{H}'[G, H] = T_1(\hat{H}[H, H_1]) = (N, \Delta, Q, A)$.
(5) $\mathscr{G}[G, H, G_1, H_1] = (M \cup N, \Delta, P \cup Q, A)$.

Again we assume that $M \cap N = \{A\}$. Clearly, $L(\mathscr{G}[G, H, G_1, H_1]) = L(\hat{G}[G, G_1]) \cup L(\hat{H}[H, H_1])$. By varying the grammars G_1 and H_1, the emptiness-of-intersection problem for s-grammars that generate languages without arbitrarily long common prefixes can be directly embedded into a variety of grammar or language class-membership problems.

PROPOSITION 4.2. *Let the fixed cfgs G_1 and H_1 be s-grammars, $M_1 \cap N_1 = \{A\}$, and the languages $L(G)$ and $L(H)$ be definite event separable. Then the cfg $\mathscr{G}[G, H, G_1, H_1]$*

(*i*) *generates an s-language, and*
(*ii*) *is simultaneously strong LL, SLR, and BRC.*

PROOF

(i) Since the languages $L(G)$ and $L(H)$ are definite event separable, so are the languages $L(\hat{G}')$ and $L(\hat{H}')$. Since the cfgs G_1 and H_1 are s-grammars, the languages $L(\hat{G}')$ and $L(\hat{H}')$ are s-languages. But $L(\mathscr{G}) = L(\hat{G}') \cup L(\hat{H}') = L(\hat{G}) \cup L(\hat{H})$. Thus, noting that every string in $L(\hat{G})$ and $L(\hat{H})$ has an endmarker, the language $L(\mathscr{G})$ is an s-language by Theorem 2.3(3).

(ii) Since the cfgs $\hat{G}$ and $\hat{H}$ are LL(1) grammars, so are the grammars $\hat{G}'$ and $\hat{H}'$. But for all nonterminals B in $M \cup N$ except A, if two productions $B \to \alpha$ and $B \to \beta$ are both elements of $P \cup Q$, then either they are both elements of P or they are both elements of Q. Since both cfgs $\hat{G}'$ and $\hat{H}'$ are LL(1) grammars, $\text{First}_1(\alpha\cdot\text{Follow}_1(B)) \cap \text{First}_1(\beta\cdot\text{Follow}_1(B)) = \varnothing$. Since the languages $L(G)$ and $L(H)$ are definite event separable, there exists an integer $k \geq 1$ such that

$$\text{First}_k(\overline{[A]}\cdot\$'\cdot\$\cdot S'\cdot \cent\cdot S_0) \cap \text{First}_k(\overline{[A]}\cdot\$''\cdot\$\cdot T'\cdot \cent\cdot T_0) = \varnothing.$$

Thus the cfg $\mathscr{G}$ is strong LL by Definition 1.5.

The cfgs $\hat{G}$ and $\hat{H}$ are strict deterministic grammars for the partitions $\pi_1 = \{\Delta\} \cup \{\{B\}\,|\,B \in \hat{M}\}$ and $\pi_2 = \{\Delta\} \cup \{\{B\}\,|\,B \in \hat{N}\}$, respectively. Hence, by the proof of [3, Th. 5.1], the cfgs $\hat{G}'$ and $\hat{H}'$ are both also strict deterministic grammars. Hence, by [7, Th. 4.1], the cfgs $\hat{G}'$ and $\hat{H}'$ are both $LR(0)$ grammars using the definition of $LR(0)$ grammar in [3] and [8]. Thus by [3, Th. 1.4], the cfgs $\hat{G}'$ and $\hat{H}'$ are both also $LR(0)$ grammars, using the definition of $LR(0)$ grammar in [1]. The only viable prefixes γ of the augmented grammar of $\mathscr{G}$ for which the set of valid $LR(0)$ items for γ contains $LR(0)$ items for both the augmented grammar of $\hat{G}'$ and the augmented grammar of $\hat{H}'$ are λ and $\overline{[A]}$. The only resulting $LR(0)$ conflicts are between the items $[\overline{[\$']} \to .\lambda, \lambda]$ and $[\overline{[\$'']} \to .\lambda, \lambda]$. Since the languages $L(\hat{G}')$ and $L(\hat{H}')$ are definite event separable, there is an integer $k \geq 1$ such that

$$\text{Follow}_k(\overline{[\$']}) \cap \text{Follow}_k(\overline{[\$'']}) = \varnothing.$$

Thus by Definition 1.6 the cfg $\mathscr{G}$ is SLR.

Finally, as noted above, the cfgs $\hat{G}$ and $\hat{H}$ are both strict deterministic grammars. Hence, by the proof of [5, Th. 5.1], the cfgs $\hat{G}'$ and $\hat{H}'$ are both BRC grammars. By inspection, any rightmost derivation of $\mathscr{G}$ is of one of the forms

(1) $A \Rightarrow^*_{\text{rm}} \overline{[A]}\cdot\overline{[\$']}\cdot\$\cdot w \Rightarrow \overline{[A]}\cdot\$\cdot w \Rightarrow \$\cdot w$, or
(2) $A \Rightarrow^*_{\text{rm}} \overline{[A]}\cdot\overline{[\$'']}\cdot\$\cdot w \Rightarrow \overline{[A]}\cdot\$\cdot w \Rightarrow \$w$,

where all derivations of the form of (1) are rightmost derivations of $\hat{G}'$ and all derivations of the form of (2) are rightmost derivations of $\hat{H}'$. Since the languages $L(\hat{G}')$ and $L(\hat{H}')$ are definite event separable, there is an integer $k \geq 1$ such that the left context $\overline{[A]}$ and at most k characters of the right context suffice to uniquely determine if a rightmost derivation of $\mathscr{G}$ is of the form of (1) or (2). Once this decision is made, the remainder of the parse can be accomplished using only productions of the cfgs $\hat{G}'$ or $\hat{H}'$, respectively. Thus, by Definition 1.7 the cfg $\mathscr{G}$ is BRC. □

Definition 4.3. We denote the set of all cfgs that are simultaneously strong LL, SLR, and BRC and that generate s-languages by $\mathscr{C}$.

THEOREM 4.4. *Let Γ be any set of cfgs containing $\mathscr{C}$ such that no inherently ambiguous cfg is a member of Γ. Then it is undecidable whether a cfg is a member of Γ.*

PROOF. Let Γ be any set of cfgs satisfying the conditions of the theorem. Let $G_1 = (\{X, Y, Z\}, \{a, b, c, d\}, P_0, X)$, where

$$P_0 = \{X \to a\cdot Y\cdot b\cdot d\cdot c\cdot Z,\ Y \to a\cdot Y\cdot b,\ Y \to d,\ Z \to c\cdot Z,\ Z \to d\}.$$

Let $H_1 = (\{U, V, W\}, \{a, b, c, d\}, Q_0, U)$, where

$$Q_0 = \{U \to a\cdot V\cdot b\cdot W\cdot c\cdot d,\ V \to a\cdot V,\ V \to d,\ W \to b\cdot W\cdot c,\ W \to d\}.$$

The cfgs G_1 and H_1 are both s-grammars. $L(G_1) = \{a^n\cdot d\cdot b^n\cdot d\cdot c^m\cdot d\,|\,n, m \geq 1\}$, and $L(H_1) = \{a^n\cdot d\cdot b^m\cdot d\cdot c^m\cdot d\,|\,n, m \geq 1\}$. Thus the language $L(G_1)\cdot\{\cent\} \cup L(H_1)\cdot\{\cent\}$ is inherently ambiguous.

Let G and H be grammars in $\mathscr{S}$ that generate languages without arbitrarily long common prefixes. Suppose $\mathscr{C} \subseteq \Gamma$. We claim that the cfg $\mathscr{G}[G, H, G_1, H_1]$ is a member of Γ if and only if $L(G) \cap L(H) = \varnothing$. Suppose $L(G) \cap L(H) = \varnothing$. Then the languages $L(G)$ and $L(H)$ are definite event separable. Hence, by Proposition 4.2 the cfg $\mathscr{G}$ is a member of $\mathscr{C}$ and thus is a member of Γ. Suppose $L(G) \cap L(H) \neq \varnothing$. Let x be any element of $L(G) \cap L(H)$. Then $\{\$\cdot x\cdot ¢\}\backslash L(\mathscr{G}) = L(G_1)\cdot\{¢\} \cup L(H_1)\cdot\{¢\}$, which as noted above is inherently ambiguous. Thus, since the unambiguous cfgs are closed under quotient on the left with a single string, the cfg $\mathscr{G}$ is inherently ambiguous and thus is not a member of Γ. Thus, by Theorem 2.3(1) the problem of determining whether a cfg is a member of Γ is not recursively time bounded and thus is undecidable. □

We extend the conclusion of Theorem 4.4 to a variety of grammatical similarity problems. To do this we need the following definition and proposition.

Definition 4.5. A cfg G is said to be

(1) *tree ambiguous* if some string x in $L(G)$ has two derivation trees that are distinct even when the labels of all nonterminal nodes are removed, and
(2) *inherently tree ambiguous* if every equivalent cfg is tree ambiguous.

PROPOSITION 4.6. *A cfg is inherently tree ambiguous if and only if it is inherently ambiguous.*

PROOF. A cfg is ambiguous if and only if it is structurally ambiguous or is tree ambiguous. It is known [22] that every cfg is equivalent to a structurally unambiguous cfg. Thus, if a cfg is equivalent to a cfg that is not tree ambiguous, it is equivalent to an unambiguous cfg. □

THEOREM 4.7. *Let ρ be any reflexive binary relation on the cfgs such that for all cfgs G and H, if $G\ \rho\ H$ and G is inherently tree ambiguous, then H is tree ambiguous. Let Γ be any class of cfgs containing $\mathscr{C}$ and containing* no *tree ambiguous cfgs. Then it is undecidable for a cfg G whether there exists a cfg H in Γ such that $G\ \rho\ H$.*

PROOF. Let ρ and Γ satisfy the conditions of the theorem. Let $\mathscr{G}_\Gamma = \{G \mid G$ is a cfg, and there exists a cfg $H \in \Gamma$ such that $G\ \rho\ H\}$. Then the set $\mathscr{G}_\Gamma$ satisfies the conditions of Theorem 4.4 and thus is not recursive. □

We present two natural and simple sufficient conditions for a binary relation ρ on the cfgs to satisfy the conditions of Theorem 4.7.

Definition 4.8. A binary relation ρ on the cfgs is said to *preserve languages* if for all cfgs G and H, if $G\ \rho\ H$, then $L(G) = L(H)$. A relation ρ is said to *preserve ambiguity* (*preserve tree ambiguity*) if for all cfgs G and H, if $G\ \rho\ H$ and G is ambiguous (tree ambiguous), then H is ambiguous (tree ambiguous).

PROPOSITION 4.9. *Let ρ be any reflexive binary relation on the cfgs that preserves languages or preserves tree ambiguity. Then the relation ρ satisfies the conditions of Theorem* 4.7.

PROOF. The proposition follows directly from the definition of inherent ambiguity and Proposition 4.6. □

Proposition 4.9 can be used to show that most of the grammatical similarity relations studied in the literature satisfy the conditions of Theorem 4.7. For example,

(1) the following relations preserve languages:

(a) $G = H$;
(b) $L(G) = L(H)$;
(c) G is structurally equivalent to H;
(d) G is λ-free left (right) covered by H;
(e) G is left (right) covered by H;
(f) G is λ-free covered by H;
(g) G is covered by H;
(h) G is isomorphic to H;

(2) the following relations preserve tree ambiguity:

(i) G is structurally contained by H;
(j) there is a homomorphism from G to H;
(k) G is Reynolds covered by H.

We conclude this section by presenting several applications of Theorem 4.7 and Proposition 4.9.

COROLLARY 4.10. *Let ρ be any of the grammatical similarity relations (a)–(k) above. Let Γ be any of the following grammar classes:*

(1) *BRC grammars;*
(2) *strong LL grammars;*
(3) *LL grammars;*
(4) *LR grammars;*
(5) *SLR grammars;*
(6) *LALR grammars;*
(7) *LL-regular grammars;*
(8) *LR-regular grammars;*
(9) *FSPA grammars;*
(10) *unambiguous grammars.*

Then it is undecidable for a cfg G whether there exists a cfg $H \in \Gamma$ such that $G \rho H$.

PROOF. Definitions and basic properties of the grammar classes (1)–(10) can be found in [1] (1)–(6), (10); [14] (7); [2] (8); and [21] (9). Each class contains $\mathscr{C}$ and contains no tree ambiguous cfgs. ☐

5. *Grammatical-Covering and Language-Class-Membership Problems*

In this section we show how the emptiness-of-intersection problem for s-grammars that generate languages without arbitrarily long common prefixes can be efficiently embedded into a variety of additional grammatical-covering and language-class-membership problems. Such efficient embeddings suffice to prove the undecidability of a variety of such problems studied in the literature.

THEOREM 5.1. *Let ρ_1 and ρ_2 be the binary relations on the cfgs defined, for all cfgs G and H, by*

(1) *$G \rho_1 H$ if and only if G is right covered by H, and*
(2) *$G \rho_2 H$ if and only if $L(G) = L(H)$.*

Let ρ be any binary relation on the cfgs between ρ_1 and ρ_2. Let Γ be any class of cfgs *that*

(*a*) *contains all LR*(0) *grammars* (*using the definition in* [1]), *and*
(*b*) *contains* no *inherently ambiguous cfgs.*

Then it is undecidable for a cfg G whether there exists a cfg $H \in \Gamma$ such that $G \rho H$.

PROOF. For any such ρ and Γ, let $\mathscr{T}[\rho, \Gamma] = \{G \mid G$ is a cfg such that there exists a cfg $H \in \Gamma$ for which $G \rho H\}$. Michunas [16] has shown that every LR grammar that generates a prefix-free language is right covered by an LR(0) grammar (using the definition in [1]). Thus $\mathscr{C} \subseteq \mathscr{T}[\rho, \Gamma]$. Since for all cfgs G and H, $G \rho H \Rightarrow L(G) = L(H)$, the set $\mathscr{T}[\rho, \Gamma]$ contains *no* inherently ambiguous cfgs. Thus by Theorem 4.4 the set $\mathscr{T}[\rho, \Gamma]$ is not recursive. □

COROLLARY 5.2. *For all $k \geq 0$ it is undecidable whether a cfg is right covered by an LR(k), SLR(k), or LALR(k) grammar.*

Analogous results hold for left covering.

THEOREM 5.3. *Let ρ_1 and ρ_2 be the binary relations on the cfgs defined, for all cfgs G and H, by*

(1) *$G \rho_1 H$ if and only if G is left covered by H, and*
(2) *$G \rho_2 H$ if and only if $L(G) = L(H)$.*

Let ρ be any binary relation on the cfgs between ρ_1 and ρ_2. Let Γ be any class of cfgs that

(*a*) *contains all cfgs that are simultaneously LL*(1) *and strict deterministic of degree one* [8] *and generate s-languages, and*
(*b*) *contains* no *inherently ambiguous cfg.*

Then it is undecidable for a cfg G whether there exists a cfg $H \in \Gamma$ such that $G \rho H$.

PROOF. Since the set of cfgs is recursive, it suffices to show that *no* such decision problem is recursively time bounded. Thus, by Theorem 2.3(1) it suffices to show that the emptiness-of-intersection problem for pairs of s-grammars generating languages without arbitrarily long common prefixes is efficiently embeddable into each such decision problem.

Let $G_1 = (M, \Delta, P_1, S_1)$ and $G_2 = (N, \Delta, P_2, S_2)$ be s-grammars generating languages without arbitrarily long common prefixes. Without loss of generality we assume that $M \cap N = \varnothing$ and that each right-hand side of a production in P_1 or P_2 has exactly one occurrence of a terminal. We also assume that $\Delta = \{0, 1\}$. We show how to construct efficiently a cfg $\mathscr{G}$ from G_1 and G_2 such that

(i) $\mathscr{G}$ is left covered by a cfg $H \in \Gamma$ if $L(G_1) \cap L(G_2) = \varnothing$, and
(ii) $\mathscr{G}$ is inherently ambiguous if $L(G_1) \cap L(G_2) \neq \varnothing$.

Thus there exists a cfg $H \in \Gamma$ such that $G \rho H$ if and only if $L(G_1) \cap L(G_2) = \varnothing$.

Let $A_1, A_2, A_3, B_1, B_2, B_3$, and S be distinct nonterminals not appearing in G_1 or in G_2. Let a, b, c, $¢$, and \$ be distinct terminals not in $\{0, 1\}$. Let $\Sigma = \{0, 1, a, b, c, ¢, \$\}$. We construct cfgs G_1', G_2', and $\mathscr{G}$ as follows.

(1) $G_1' = (M', \Sigma, P, S)$, where

(i) $M' = M \cup \{A_1, A_2, A_3, S\}$, and
(ii) $P = P_1 \cup \{S \to ¢ \cdot S_1 \cdot ¢ \cdot A_1, A_1 \to a \cdot A_2 \cdot b \cdot ¢ \cdot c \cdot A_3, A_2 \to a \cdot A_2 \cdot b, A_2 \to ¢, A_3 \to c \cdot A_3, A_3 \to \$\}$;

(2) $G_2' = (N', \Sigma, Q, S)$, where

(i) $N' = N \cup \{B_1, B_2, B_3, S\}$, and
(ii) $Q = P_2 \cup \{S \to ¢\cdot S_2 \cdot ¢ \cdot B_1,\ B_1 \to a\cdot B_2 \cdot b \cdot B_3 \cdot c \cdot \$,\ B_2 \to a \cdot B_2,\ B_2 \to ¢,$ $B_3 \to b \cdot B_3 \cdot c,\ B_3 \to ¢\}$; and

(3) $\mathscr{G} = (V, \Sigma, \Pi, S)$, where

(i) $V = M' \cup N'$, and
(ii) $\Pi = P \cup Q$.

By inspection the cfgs G_1' and G_2' are s-grammars, $L(G_1') = \{¢\} \cdot L(G_1) \cdot \{¢ \cdot a^n \cdot ¢ \cdot b^n \cdot ¢ \cdot c^m \cdot \$ \mid n,\ m \geq 1\}$, $L(G_2') = \{¢\} \cdot L(G_2) \cdot \{¢ \cdot a^n \cdot ¢ \cdot b^m \cdot ¢ \cdot c^m \cdot \$ \mid n,\ m \geq 1\}$, and $L(\mathscr{G}) = L(G_1') \cup L(G_2')$.

Case 1. Suppose $L(G_1) \cap L(G_2) \neq \varnothing$. Let $x \in L(G_1) \cap L(G_2)$. Then the language $\{¢ \cdot x \cdot ¢\} \backslash L(\mathscr{G})$ is $\{a^i \cdot ¢ \cdot b^j \cdot ¢ \cdot c^k \cdot \$ \mid i,\ j,\ k \geq 1$, and $i = j$ or $j = k\}$ and thus is inherently ambiguous. Since the unambiguous cfls are closed under quotient on the left with single string, the cfg $\mathscr{G}$ is also inherently ambiguous. Thus, by (2) and (b) there is no grammar $H \in \Gamma$ such that $\mathscr{G}\ \rho_2\ H$. Hence there is no grammar $H \in \Gamma$ such that $\mathscr{G}\ \rho\ H$.

Case 2. Suppose $L(G_1) \cap L(G_2) = \varnothing$. Then $L(G_1') \cap L(G_2') = \varnothing$, and the s-grammars G_1' and G_2' also generate languages without arbitrarily long common prefixes. Thus, by Theorem 2.3(3) the language $L(\mathscr{G})$ is an s-language. A cfg $\mathscr{G}'$ and a homomorphism h such that

(i) $\mathscr{G}$ is left covered by $\mathscr{G}'$ by h, and
(ii) $\mathscr{G}'$ is simultaneously an LL(1) and a strict deterministic grammar of degree 1

is constructed in Appendix B. The idea of this construction is that $\mathscr{G}'$, for any $x \in L(\mathscr{G}')$, simultaneously keeps track of the leftmost derivation of x by G_1' and of x by G_2' until the prefix of x generated suffices to determine if $x \in L(G_1')$ or $x \in L(G_2')$. Subsequently the grammar simulates the remainder of the derivation of x by G_1' or G_2', respectively. (This is possible since the languages $L(G_1')$ and $L(G_2')$ do *not* have arbitarily long common prefixes and $L(G_1') \cap L(G_2') = \varnothing$.) Thus, by (1) and (a) there exists a cfg $H \in \Gamma$ such that $\mathscr{G}\ \rho_1\ H$. Hence there exists a cfg $H \in \Gamma$ such that $\mathscr{G}\ \rho\ H$. □

COROLLARY 5.4. *It is undecidable whether a cfg is left covered by*

(1) *an $LL(k)$, strong $LL(k)$, $LR(k)$, $SLR(k)$ or $LALR(k)$ grammar, for all $k \geq 1$;*
(2) *a strict deterministic grammar of degree* 1;
(3) *a strict deterministic grammar;*
(4) *an $LR(0)$ grammar;*
(5) *an LL, strong LL, LR, SLR, or LALR grammar; or*
(6) *an unambiguous cfg.*

PROOF. Each such grammar class satisfies conditions (1) and (2) of Theorem 5.3. □

6. *Conclusion*

We have studied the decidability of context-free grammar and language problems in general, especially problems about separability and context, grammar and language class membership, and grammatical similarity relations. Our results are based upon efficient reductions of membership problems for always-halting Turing machines and the following new fundamental property of s-grammars and s-languages:

> The emptiness-of-intersection problem for pairs of s-grammars that generate languages *without* arbitrarily long common prefixes is decidable but is *not* recursively time bounded.

This property implies much, if not most, of what was previously known about the undecidability of context-free grammar problems, and implies many new results as well, for example, Theorems 3.1, 4.4, 4.7, 5.1, and 5.3. It can also be used to prove analogous complexity and/or undecidability results for various proper subclasses of the cfgs, including the LL, LR, LL-regular, LR-regular, FSPA(1), LR(1, ∞), and unambiguous grammars (see [10]).

Appendix A

Let M be a single-tape deterministic Tm with state set Q, tape alphabet T, start state q_0, and final state q_f such that M always adds new tape squares on the right end of its tape. Let the set of allowable moves of M be δ_M, where δ_M is some finite subset of $Q \times T \times Q \times T \times \{L, R, S\}$, where L, R, and S denote left, right, and stationary, respectively. Let $\Sigma = T \cup \hat{T} \cup (Q \times T) \cup (Q \times \hat{T}) \cup \{\#, \$\}$, where $\hat{T} = \{\hat{b} \mid b \in T\}$, and we assume that the sets T, $\hat{T}$, $(Q \times T)$, $(Q \times \hat{T})$, and $\{\#, \$\}$, are pairwise disjoint.

The s-grammar $G_1[M, x]$ is defined by $G_1[M, x] = (\{S_1, A_1, A_2, A_3, B\}, \Sigma, P_1, S_1)$, where the productions of $G_1[M, x]$ are

(1) $S_1 \to \#\cdot(q_0, x_1)\cdot x_2 \cdot \cdots \cdot x_{n-1}\cdot\hat{x}_n\cdot\#\cdot A_1$,
(2a) $A_1 \to \hat{b}\cdot A_2\cdot\hat{b}\cdot\#\cdot A_1$, for all $b \in T$,
(2b) $A_1 \to (s, \hat{b})\cdot A_3\cdot(s, \hat{b})\cdot\#\cdot A_1$, for all $(s, \hat{b}) \in (Q - \{q_f\}) \times \hat{T}$,
(2c) $A_1 \to (q_f, \hat{b})\cdot B$, for all $b \in T$,
(3a) $A_2 \to b\cdot A_2\cdot b$, for all $b \in T$,
(3b) $A_2 \to (s, b)\cdot A_3\cdot(s, b)$, for all $(s, b) \in (Q - \{q_f\}) \times T$,
(4a) $A_3 \to b\cdot A_3\cdot b$, for all $b \in T$,
(4b) $A_3 \to \#$,
(5a) $B \to b\cdot B$, for all $b \in T$,
(5b) $B \to \#\cdot\$$.

The s-grammar $G_2[M, x]$ is defined by $G_2[M, x] = (V, \Sigma, P_2, S_2)$, where

$$\begin{aligned} V = {} & \{S_1, S_2\} \cup \{A\langle a, b\rangle \mid a \in T \cup \{\#\} \text{ and } b \in T\} \\ & \cup \{A^1\langle a, (q, b)\rangle \mid a \in T \cup \{\#\} \text{ and } (q, b) \in Q \times T\} \\ & \cup \{A^2\langle (q, b), a\rangle \mid (q, b) \in Q \times T \text{ and } a \in T\} \\ & \cup \{A^3\langle a, b\rangle \mid a, b \in T\}, \end{aligned}$$

and the productions of $G_2[M, x]$ are

(1) $S_1 \to \#\cdot S_2$,
(2a) $S_2 \to a\cdot A\langle\#, a\rangle\cdot\#\cdot S_2$, for all $a \in T$,
(2b) $S_2 \to (q, b)\cdot A^1\langle\#, (q, b)\rangle\cdot\#\cdot S_2$, for all $(q, b) \in Q \times T$,
(2c) $S_2 \to \$$,
(3a) $A\langle\#, a\rangle \to b\cdot A\langle a, b\rangle\cdot a$, for all $a, b \in T$,
(3b) $A\langle\#, a\rangle \to (q, b)\cdot A^1\langle a, (q, b)\rangle\cdot a$, for all $a \in T$ and all $(q, b) \in Q \times T$ such that $(q, b, -, -, R)$ or $(q, b, -, -, S) \in \delta_M$,
(3c) $A\langle\#, a\rangle \to (q, b)\cdot A^1\langle a, (q, b)\rangle\cdot(r, a)$, for all $a \in T$ and $(q, b) \in Q \times T$ such that $(q, b, r, -, L) \in \delta_M$,

(4a) $A^1\langle \#, (q, b)\rangle \rightarrow a\cdot A^2\langle (q, b), a\rangle \cdot (r, d)$, for all $a, b \in T$, if $(q, b, r, d, S) \in \delta_M$,
(4b) $A^1\langle \#, (q, b)\rangle \rightarrow a\cdot A^2\langle (q, b), a\rangle \cdot d$, for all $a, b \in T$, if $(q, b, —, d, R) \in \delta_M$.

(No rule of the form (q, b, r, d, L) is possible here, since by assumption M always adds new tape squares on the right end of its tape.)

(5a) $A\langle a, b\rangle \rightarrow c\cdot A\langle b, c\rangle \cdot b$, for all $a, b, c \in T$,
(5b) $A\langle a, b\rangle \rightarrow (q, c)\cdot A^1\langle b, (q, c)\rangle \cdot (r, b)$, for all $a, b \in T$ and all $(q, c) \in Q \times T$ such that $(q, c, r, —, L) \in \delta_M$,
(5c) $A\langle a, b\rangle \rightarrow (q, c)\cdot A^1\langle b, (q, c)\rangle \cdot b$, for all $a, b, \in T$ and all $(q, c) \in Q \times T$ such that $(q, c, —, —, S)$ and $(q, c, —, —, R) \in \delta_M$,
(5d) $A\langle a, b\rangle \rightarrow (q, \hat{c})\cdot \# \cdot \hat{c}\cdot (r, b)$, for all $a, b \in T$ and all $(q, \hat{c}) \in Q \times \hat{T}$ such that $(q, c, r, —, L) \in \delta_M$,
(5e) $A\langle a, b\rangle \rightarrow (q, \hat{c})\cdot \# \cdot (r, \hat{d})\cdot b$, for all $a, b \in T$ and all $(q, \hat{c}) \in Q \times \hat{T}$ such that $(q, c, r, d, S) \in \delta_M$,
(5f) $A\langle a, b\rangle \rightarrow (q, \hat{c})\cdot \# \cdot (r, \hat{\not{b}})\cdot d\cdot b$, for all $a, b \in T$ and all $(q, \hat{c}) \in Q \times \hat{T}$ such that $(q, c, r, d, R) \in \delta_M$.

(We denote the blank tape symbol by $\not{b}$.)

(6a) $A^1\langle a, (q, b)\rangle \rightarrow c\cdot A^2\langle (q, b), c\rangle \cdot d$, for all $a, c \in T$ and all $(q, b) \in Q \times T$, if $(q, b, —, d, L)$ or $(q, b, —, d, R) \in \delta_M$,
(6b) $A^1\langle a, (q, b)\rangle \rightarrow c\cdot A^2\langle (q, b), c\rangle \cdot (r, d)$, for all $a, c \in T$ and all $(q, b) \in Q \times T$, if $(q, b, r, d, S) \in \delta_M$,
(6c) $A^1\langle a, (q, b)\rangle \rightarrow \hat{c}\cdot \# \cdot \hat{c}\cdot d$, for all $a \in T$, all $\hat{c} \in \hat{T}$, and all $(q, b) \in Q \times T$, if $(q, b, —, d, L) \in \delta_M$,
(6d) $A^1\langle a, (q, b)\rangle \rightarrow \hat{c}\cdot \# \cdot \hat{c}\cdot (r, d)$, for all $a \in T$, all $\hat{c} \in \hat{T}$, and all $(q, b) \in Q \times T$, if $(q, b, r, d, S) \in \delta_M$.

Appendix B

Let $G_1 = (M, \Delta, P_1, S_1)$, $G_2 = (N, \Delta, P_2, S_2)$, $G_1' = (M', \Sigma, P, S)$, $G_2' = (N', \Sigma, Q, S)$ and $\mathscr{G} = (V, \Sigma, \Pi, S)$ be defined as in the proof of Theorem 4.2. Let $|P| = m$ and $|Q| = n$. Let

$$\begin{aligned}\mathscr{V} = \{(S_1; S_2)\} \cup \{(\alpha, i_1, \ldots, i_k; \beta, j_1, \ldots, j_k) \mid {}& \alpha \in M^*, \beta \in N^*, \\ & 1 \le i_1, \ldots, i_k \le m, 1 \le j_1, \ldots, j_k \le n, \text{ and there exists a} \\ & \text{nonempty terminal string } x \text{ such that } S_1 \Rightarrow_{G_1,\mathrm{lm}}^{i_1\cdots i_k} x\cdot\alpha \text{ and} \\ & S_2 \Rightarrow_{G_2,\mathrm{lm}}^{j_1\cdots j_k} x\cdot\beta\}.\end{aligned}$$

Let $C_0, C_1, \ldots, C_m, D_0, D_1, \ldots, D_n$ be distinct nonterminals not in V or $\mathscr{V}$. Then $\mathscr{G}' = (V', \Sigma, \Pi', S)$, where

(1) $V' = V \cup \mathscr{V} \cup \{C_0, C_1, \ldots, C_m, D_0, D_1, \ldots, D_n\}$; and
(2) Π' is the union of the sets,

(a) $\{S \rightarrow \text{¢}\cdot(S_1; S_2)\}$,
(b) $\{(\alpha, i_1, \ldots, i_k; \beta, j_1, \ldots, j_k) \rightarrow a\cdot(\alpha', i_1, \ldots, i_{k+1}; \beta', j_1, \ldots, j_{k+1}) \mid$ for all $a \in \{0, 1\}$ and all $(\alpha, i_1, \ldots, i_k; \beta, j_1, \ldots, j_k) \in \mathscr{V}$ for which $\alpha \rightarrow_{G_1',\mathrm{lm}}^{i_{k+1}} a\cdot\alpha'$ and $\beta \rightarrow_{G_2',\mathrm{lm}}^{j_{k+1}} a\cdot\beta'\}$,
(c) $\{(\alpha, i_1, \ldots, i_k; \beta, j_1, \ldots, j_k) \rightarrow a\cdot C_0\cdot C_{i_1}\cdot \cdots \cdot C_{i_{k+1}}\cdot\alpha'\cdot\text{¢}\cdot A_1 \mid$ for all $a \in \{0, 1\}$ and all $(\alpha, i_1, \ldots, i_k; \beta, j_1, \ldots, j_k) \in \mathscr{V}$ for which $\alpha \rightarrow_{G_1',\mathrm{lm}}^{i_{k+1}} a\cdot\alpha'$ and $\beta \rightarrow_{G_2',\mathrm{lm}} b\cdot\beta'$ implies $b \ne a\}$,
(d) $\{(\alpha, i_1, \ldots, i_k; \beta, j_1, \ldots, j_k) \rightarrow a\cdot D_0\cdot D_{j_1}\cdot \cdots \cdot D_{j_{k+1}}\cdot\beta'\cdot\text{¢}\cdot B_1 \mid$ for all $a \in$

$\{0, 1\}$ and all $(\alpha, i_1, \ldots, i_k; \beta, j_1, \ldots, j_k) \in \mathscr{V}$ for which $\beta \rightarrow^{j_{k+1}}_{G_{2,\mathrm{lm}}'} a \cdot \beta'$ and $\alpha \rightarrow_{G_{1,\mathrm{lm}}'} b \cdot \alpha'$ implies $b \neq a\}$,

(e) $\{(\alpha, i_1, \ldots, i_k; \lambda, j_1, \ldots, j_k) \rightarrow a \cdot C_0 \cdot C_{i_1} \cdot \cdots \cdot C_{i_{k+1}} \cdot \alpha' \cdot ¢ \cdot A_1 \mid$ for all $a \in \{0, 1\}$ and all $(\alpha, i_1, \ldots, i_k; \lambda, j_1, \ldots, j_k) \in \mathscr{V}$ for which $\alpha \rightarrow^{i_{k+1}}_{G_{1,\mathrm{lm}}'} a \cdot \alpha'\}$,

(f) $\{(\lambda, i_1, \ldots, i_k; \beta, j_1, \ldots, j_k) \rightarrow a \cdot D_0 \cdot D_{j_1} \cdot \cdots \cdot D_{j_{k+1}} \cdot \beta' \cdot ¢ \cdot B_1 \mid$ for all $a \in \{0, 1\}$ and all $(\lambda, i_1, \ldots, i_k; \beta, j_1, \ldots, j_k) \in \mathscr{V}$ for which $\beta \rightarrow^{j_{k+1}}_{G_{2,\mathrm{lm}}'} a \cdot \beta'\}$,

(g) $\{(\alpha, i_1, \ldots, i_k; \lambda, j_1, \ldots, j_k) \rightarrow ¢ \cdot D_0 \cdot D_{j_1} \cdot \cdots \cdot D_{j_{k+1}} \cdot B_1 \mid$ for all $(\alpha, i_1, \ldots, i_k; \lambda, j_1, \ldots, j_k) \in \mathscr{V}\}$,

(h) $\{(\lambda, i_1, \ldots, i_k; \beta, j_1, \ldots, j_k) \rightarrow ¢ \cdot C_0 \cdot C_{i_1} \cdot \cdots \cdot C_{i_{k+1}} \cdot A_1 \mid$ for all $(\lambda, i_1, \ldots, i_k; \beta, j_1, \ldots, j_k) \in \mathscr{V}\}$,

(i) $P - \{S \rightarrow ¢ \cdot S_1 \cdot ¢ \cdot A_1\}$,

(j) $Q - \{S \rightarrow ¢ \cdot T_1 \cdot ¢ \cdot B_1\}$,

(k) $C_i \rightarrow \lambda$, for $0 \leq i \leq m$, and

(l) $D_j \rightarrow \lambda$, for $0 \leq j \leq n$.

Let h be the homomorphism from Π' to $\Pi \cup \{\lambda\}$ defined by

(1) for all productions ρ in sets (a) through (h), $h(\rho) = \lambda$;
(2) for all productions ρ in sets (i) and (j), $h(\rho) = \rho$;
(3) the production $h(C_0 \rightarrow \lambda)$ is $S \rightarrow ¢ \cdot S_1 \cdot ¢ \cdot A_1$;
(4) the production $h(D_0 \rightarrow \lambda)$ is $S \rightarrow ¢ \cdot S_2 \cdot ¢ \cdot B_1$,
(5) for all i with $1 \leq i \leq m$, the production $h(C_i \rightarrow \lambda)$ is the ith production of G_1, and
(6) for all j with $1 \leq j \leq n$, the production $h(D_j \rightarrow \lambda)$ is the jth production of G_2.

We claim that $\mathscr{G}$ is left covered by $\mathscr{G}'$ by the homomorphism h and that $\mathscr{G}'$ is simultaneously an LL(1) and a strict deterministic grammar of degree 1.

Since $\mathscr{G}$ is unambiguous, to show that $\mathscr{G}'$ left covers $\mathscr{G}$, it suffices to show that $L(\mathscr{G}') = L(\mathscr{G})$ and that

($*$) for all leftmost derivations Π of $\mathscr{G}'$, if $S \Rightarrow^{\Pi}_{\mathscr{G}',\mathrm{lm}} w$, then $S \Rightarrow^{h(\Pi)}_{\mathscr{G},\mathrm{lm}} w$.

Since each of the nonterminals $C_0, C_1, \ldots, C_m, D_0, D_1, \ldots, D_n$ only generates λ, $L(\mathscr{G}') = L(\mathscr{G})$. The proof of ($*$) follows by induction on the length of Π, noting that the nonterminals in $\mathscr{V}$ contain all information needed to construct the corresponding leftmost derivation of $\mathscr{G}$. Let $w \in L(\mathscr{G})$. The homomorphism h, when applied to the productions $C_0 \rightarrow \lambda, C_{i_1} \rightarrow \lambda, \ldots, C_{i_{k+1}} \rightarrow \lambda$ or $D_0 \rightarrow \lambda, D_{j_1} \rightarrow \lambda, \ldots, D_{j_{k+1}} \rightarrow \lambda$ in w's leftmost derivation by $\mathscr{G}'$, is used to synchronize the leftmost derivation of w by $\mathscr{G}'$ with the corresponding leftmost derivation of w by $\mathscr{G}$ once the prefix of w generated suffices to determine whether w is in $L(G_1')$ or $L(G_2')$. Since the languages $L(G_1')$ and $L(G_2')$ do not have arbitrarily long comon prefixes and $L(G_1') \cap L(G_2') = \varnothing$, this is always possible. Thus $\mathscr{G}$ is left covered by $\mathscr{G}'$ by the homomorphism h.

Finally, by inspection of $\mathscr{G}'$, for all $A \in V'$, if $A \rightarrow \alpha$ and $A \rightarrow \beta$ are distinct productions in Π', then there exist distinct terminals a and b in Σ such that $\alpha = a \cdot \alpha'$ and $\beta = b \cdot \beta'$. Thus, by [1, Th. 5.3] and [7, Th. 3.1], $\mathscr{G}'$ is simultaneously an LL(1) and a strict deterministic grammar of degree 1.

REFERENCES

(Note. References [15, 23] are not cited in the text.)

1. AHO, A.V., AND ULLMAN, J D *The Theory of Parsing, Translation, and Compiling, Vols. 1, 2* Prentice-Hall, Englewood Cliffs, N.J., 1972, 1973.
2. CULIK, K. II, AND COHEN, R LR-regular grammars—An extension of LR(k) grammars. *J. Comput. Syst. Sci. 7* (1973), 66–96.
3. GELLER, M M., AND HARRISON, M.A. Characterizations of LR(0) languages. Conf. Rec 14th Ann. Symp. on Switching and Automata Theory, Ames, Iowa, Oct. 1973, pp. 103–108

4. GRAY, J.N., AND HARRISON, M.A. On the covering and reduction problems for context-free grammars. *J. ACM 19*, 4 (Oct. 1972), 675–698
5. HARRISON, M.A., AND HAVEL, I.M. On the parsing of deterministic languages. *J. ACM 21*, 4 (Oct. 1974), 525–548
6. HARTMANIS, J. Context-free languages and Turing machine computations. In *Proceedings of a Symposium on Applied Mathematics*, Mathematical Aspects of Computer Science 19, American Mathematical Society, Providence, R.I., 1967, pp. 42–51.
7. HARTMANIS, J., AND HOPCROFT, J E. An overview of the theory of computational complexity. *J. ACM 18*, 3 (July 1971), 444–475.
8 HAVEL, I.M., AND HARRISON, M.A. Real-time strict deterministic languages. *SIAM J. Comput. 1* (1972), 333–349
9. HOPCROFT, J.E , AND ULLMAN, J.D. *Formal Languages and Their Relation to Automata.* Addison-Wesley, Reading, Mass., 1969.
10. HUNT, H.B. III. Partial set recognition problems, terminating Turing machine computations, and the complexity of correspondence problems, grammars, and program schemes. Unpublished manuscript, 1980.
11. HUNT, H B. III, AND SZYMANSKI, T.G. Dichotomization, reachability, and the forbidden subgraph property Proc. 8th Ann. ACM Symp. on Theory of Computing, Hershey, Pa., May 1976, pp. 126–134.
12 HUNT, H.B. III, AND SZYMANSKI, T.G. Complexity metatheorems for context-free grammar problems. *J. Comput. Syst. Sci. 13* (1976), 318–334.
13. HUNT, H.B. III, AND SZYMANSKI, T.G. Lower bounds and reductions between grammar problems, *J. ACM 25*, 1 (Jan. 1978), 32–51.
14 JARZABEK, S., AND KRAWCZYK, T. LL-regular grammars *Inf Proc. Lett. 4* (1975), 31–37.
15. LEWIS, P.M. II, ROSENKRANTZ, D.J., AND STEARNS, R.E. *Compiler Design Theory.* Addison-Wesley, Reading, Mass., 1976.
16. MICKUNAS, M D. On the complete covering problem for LR(k) grammars. *J. ACM 23*, 1 (Jan. 1976), 17–30.
17. OGDEN, W.F. Unpublished memorandum, December 1971.
18. PAULL, M.C., AND UNGER, S.H. Structural equivalence of context-free grammars. *J. Comput. Syst. Sci. 2* (1968), 427–463
19. REYNOLDS, J.C , AND HASKELL, R. Grammatical coverings. Unpublished manuscript, 1970.
20 SZYMANSKI, T G. Generalized bottom-up parsing. Ph D. Dissertation, Dep. of Computer Science, Cornell Univ , Ithaca, N.Y., 1973
21. SZYMANSKI, T.G., AND WILLIAMS, J.H. Noncanonical extensions of bottom-up parsing techniques. *SIAM J. Comput. 5* (1976), 231–250.
22. THATCHER, J W. Tree automata: An informal survey. In *Currents in the Theory of Computing*, A V. Aho, Ed., Prentice-Hall, Englewood Cliffs, N.J., 1973, pp. 143–172.
23. VALIANT, L.G. A note on the succinctness of descriptions of deterministic languages. *Inf. Control 32* (1976), 139–145.

RECEIVED NOVEMBER 1978; REVISED DECEMBER 1980; ACCEPTED JANUARY 1981