INFORMATION AND CONTROL 66, 103–121 (1985)

The Complexity of Equivalence Problems for Commutative Grammars

DUNG T. HUYNH

Computer Science Department, Iowa State University, Ames, Iowa 50011

In this paper we investigate the computational complexity of the inequivalence problems for commutative grammars. We show that the inequivalence problems for type 0 and context-sensitive commutative grammars are undecidable whereas decidability in nondeterministic exponential-time holds for the classes of regular and context-free commutative grammars. For the latter the inequivalence problems are Σ_2^p-hard. © 1985 Academic Press, Inc.

0. INTRODUCTION

In this paper we continue our investigation of the computational complexity of commutative grammars that have been introduced in [6]. We will focus our attention on the complexity of the equivalence problems for various commutative grammar classes. (The word problems have been investigated in [6].)

Among others we will show the following results:

Concerning the classes of type 0 and context-sensitive commutative grammars the inequivalence problems are undecidable. The proof employs Rabin's proof for vector addition systems presented in [3] by Hack.

We will show that the inequivalence problems for context-free and regular grammars are decidable in nondeterministic exponential time and that they are Σ_2^p-hard. For the finite case we obtain a sharper result. We show that the finite inequivalence problems for these classes are Σ_2^p-complete. Although a sharp result is not obtained for the general case, we will see that in the case of 1-letter terminal alphabet these problems are Σ_2^p-complete.

The paper is organized as follows. There are three sections. Section 1 deals with type 0 and context-sensitive commutative grammars. Section 2 is devoted to context-free commutative grammars. In Section 3 we investigate the complexity of inequivalence problems for regular commutative grammars and rational expressions.

The reader is referred to [6, Sect. 1] for definitions and notations. We reproduce here only a few important ones.

0019-9958/85 $3.00

Let V be a finite alphabet. V^* denotes the free monoid generated by V. ε denotes the empty word and $V^+ := V^* \backslash \{\varepsilon\}$. We shall use $V^\oplus$ to denote the free commutative monoid generated by V. If $V = \{v_1, ..., v_r\}$, then a word w in $V^\oplus$ will be written in the form

$$w = v_1^{i_1} \cdots v_r^{i_r}, \qquad i_j \in \mathbb{N}_0, \qquad j = 1, ..., r,$$

where $\mathbb{N}_0$ denotes the set of nonnegative integers. Thus w with $i_j = 0$, $j = 1, ..., r$, is the empty word of $V^\oplus$ and is also denoted by ε. A word in $V^\oplus$ is sometimes called a commutative word. $V_+^\oplus$ denotes the free commutative semigroup generated by V: $V_+^\oplus = V^\oplus \backslash \{\varepsilon\}$. In $V^\oplus$ concatenation is sometimes written as addition, e.g., $w = u + v$, where $u, v, w \in V^\oplus$.

We define a homomorphism from V^* into $V^\oplus$ as follows. Again let $V = \{v_1, ..., v_r\}$. For $j = 1, ..., r$ let $\#(v_j, w)$ denote the number of occurrences of v_j in w, where w is in V^*. Define

$$\psi_V\colon V^* \to V^\oplus$$
$$w \to v_1^{\#(v_1, w)} \cdots v_r^{\#(v_r, w)}.$$

ψ_V is known as the Parikh mapping on V^*.

Definition 0.1. A 4-tuple $G^c = (N, T, S, P^c)$ is called a *commutative* (com. for short) *grammar* iff the following conditions hold:

(1) N and T are disjoint finite alphabets,

(2) $S \in N$,

(3) P^c is a finite subset of $N_+^\oplus \times (N \cup T)^\oplus$.

As usual, N is the set of nonterminals, T is the set of terminals, S is the axiom and P^c is the set of productions.

The language generated by G^c, denoted by $L(G^c)$, is

$$L(G^c) := \{w \in T^\oplus : S \overset{*}{\Rightarrow} w\} \subset T^\oplus.$$

Definition 0.2. Let $G^c = (N, T, S, P^c)$ be a com. grammar. G^c is said to be

(1) of *type* 0 if there is no restriction on P^c.

(2) *context-sensitive* (c.s. for short) if for each $p = (\gamma, \delta) \in P^c$ it holds[1] $|\delta| \geqslant |\gamma|$.

[1] For $w \in V^*$, $|w|$ denotes the length of w and for $\tilde{w} \in V^\oplus$, $|\tilde{w}|$ also denotes the length of $\tilde{w}$ which is the sum of the exponents of $\tilde{w}$ written as com. word.

(3) *context-free* (c.f. for short) if P^c is a subset of $N \times V^{\oplus}$, i.e., each production has the form (A, δ), where $A \in N$.

(4) *regular* (reg. for short) if P^c is a subset of $N \times (T^{\oplus} \cdot (N \cup \{\varepsilon\}))$, i.e., each production is of the form (A, xB), $x \in T^{\oplus}$, $A \in N$, and $B \in N \cup \{\varepsilon\}$.

A com. language $L \subset T^{\oplus}$ is said to be of type 0 (c.s., c.f., reg.) if there is a type 0 (c.s., c.f., reg.) com. grammar G^c such that $L(G^c) = L$.

Definition 0.3. (1) The size of the grammar $G = (N, T, S, P)$, denoted by $\|G\|$, is the following number:

$$\|G\| := \log(\# V) \cdot \left(\sum_{(\gamma, \delta) \in P} |\gamma| + |\delta| \right),$$

where $V := N \cup T$. (All logarithms are to the base 2.)

(2) For a com. word $w \in V^{\oplus}$, $w = v_1^{e_1} \cdots v_r^{e_r}$, $e_j \in \mathbb{N}_0$, $j = 1, ..., r$, let $\exp(w)$ denote the number

$$\exp(w) := \sum_{j=1, e_j \geqslant 1}^{r} \lceil \log(e_j) \rceil.$$

The size of w, denoted by $\|w\|$, is defined as

$$\|w\| := \log(\# V) \cdot \exp(w).$$

(3) The size of the com. grammar $G^c = (N, T, S, P^c)$ is

$$\|G^c\| := \log(\# V) \cdot \left(\sum_{(\gamma, \delta) \in P^c} \exp(\gamma) + \exp(\delta) \right),$$

where $V := N \cup T$.

Definition 0.4. Let $\mathscr{G}$ be a class of com. grammars. The inequivalence problem for $\mathscr{G}$ is: Given two com. grammars G_1^c and G_2^c in $\mathscr{G}$ it is to determine whether $L(G_1^c) \neq L(G_2^c)$. The finite inequivalence problem for $\mathscr{G}$ is: Given two com. grammars G_1^c and G_2^c in $\mathscr{G}$ generating finite languages it is to determine whether $L(G_1^c) \neq L(G_2^c)$.

1. Complexity of the Equivalence Problems for Type 0 and Context-Sensitive Commutative Grammars

In this section we show that the inequivalence problems for type 0 and context-sensitive (c.s.) com. grammars are undecidable. Let INEQ-CSCG denote the inequivalence problem for c.s. com. grammars. We modify the proof of the undecidability of the inequivalence problem for vector addition systems presented in [3] so that we obtain a recursive reduction from Hilbert's tenth problem to INEQ-CSCG, showing that INEQ-CSCG is undecidable.

We first define the polynomial graph inclusion problem which is undecidable and then reduce this to INEQ-CSCG.

We introduce some notions. A polynomial $Q(U_1,..., U_n)$ in the polynomial ring $\mathbb{Z}[U_1,..., U_n]$ is called diophantine. ($\mathbb{Z}$ is the ring of integers.) Hilbert's tenth problem (HP) is the problem of deciding whether a diophantine polynomial has an integer solution. It is well known that HP is undecidable.

Consider polynomials in $\mathbb{N}_0[U_1,..., U_n]$. The graph of a polynomial $Q(U_1,..., U_n)$ in $\mathbb{N}_0[U_1,..., U_n]$ is the set (cf. [3])

$$G(Q)=\{(a_1,..., a_n, b)\in \mathbb{N}_0^{n+1} \mid b\leqslant Q(a_1,..., a_n)\}.$$

The polynomial graph inclusion problem (PGIP for short) is the problem of deciding for two polynomials Q_1, Q_2 in $\mathbb{N}_0[U_1,..., U_n]$ whether $G(Q_1)\subseteq G(Q_2)$.

LEMMA 1.1. *PGIP is undecidable.*

Proof. See [3] for a reduction of *HP* to *PGIP.* ▌

We now show how to reduce PGIP to INEQ-CSCG. The proof technique is essentially similar to the one presented in [3]. Instead of simulating addition and multiplication of integers by weak Petri net computers we simulate these operations by c.s. com. productions. Since c.s. com. productions are in restricted form, we do not have the freedom as in the general case. (Note that we do not have ε-productions in c.s. com. grammars.)

In the sequel we first show how to "weakly compute" a monomial by c.s. com. productions. (We use the term "weak computation" as in the case of Petri nets because $G(P)$ is, strictly speaking, not the graph of P.)

Weak Computation of a Monomial

Let $M=M(U_1,..., U_n)$ be a monomial. Then the graph $G(M)$ of M is the set $\{(a_1,..., a_n, b)\in \mathbb{N}_0^{n+1} \mid b\leqslant M(a_1,..., a_n)\}$. Let $X_1,..., X_n$ be symbols and \$, ¢ be two special symbols.[2] Further let $X'_1,..., X'_n$, Z be other symbols. We want to construct a set P of c.s. com. productions such that $\$X_1^{a_1}\cdots X_n^{a_n} \Rightarrow_P^* ¢w$ with $w\in\{Z, X'_1,..., X'_n\}^{\oplus}$ iff $w=Z^b X_1'^{a_1}\cdots X_n'^{a_n}$, where $b\leqslant M(a_1,..., a_n)$. Such a set of c.s. com. productions is said to weakly com-

[2] Special symbols are control symbols which start or stop computations. (This will be clear later.)

pute the monomial M. In order to construct such a set P, we need to consider addition and multiplication of integers.

We introduce some technical notions. Let X, Y, Z, X', Y' be some symbols and \$, ¢ be two special symbols. Let P be a set of c.s. com. productions such that $\$X^eY^f \Rightarrow_P^* \text{¢}w$ with $w \in \{Z, X', Y'\}^\oplus$ iff $w = Z^iX'^eY'^f$, where $i, e, f, \in \mathbb{N}_0$ and $i \leqslant ef$. We say that P weakly computes the product of nonnegative integers, X^e and Y^f are arguments of the weak computation by P. Further, they are reproduced as X'^e, Y'^f. We say that the weak computation of P does not consume its arguments. Similar definitions hold for weak computations of other operations.

The following lemma shows that the product of two nonnegative integers can be weakly computed.

Lemma 1.2. *Let X, Y, Z be some symbols and* \$, ¢ be two special symbols. *Further let X', Y' be some other symbols. Then there is a set P of c.s. com. productions such that $\$X^eY^f \Rightarrow_P^* \text{¢}w$* with $w \in \{Z, X', Y'\}^\oplus$ *iff $w = \text{¢}Z^iX'^eY'^f$, where $i, e, f \in \mathbb{N}_0$ and $i \leqslant ef$.*

Proof. Define P to be the following set of productions:

(1) $\$ \to A_1 \mid \text{¢},\ XA_1 \to X'A_2,\ YA_2 \to ZA_3A_4,\ A_3 \to A_2,$

(2) $A_2 \to B_1,\ B_1A_4 \to YB_1,\ B_1 \to A_1,$

(3) $A_2 \to C_1,\ C_1A_4 \to Y'C_1,\ C_1 \to \text{¢},$

(4) $X\text{¢} \to X'\text{¢},\ Y\text{¢} \to Y'\text{¢},\ A_4\text{¢} \to Y'\text{¢},$

where $A_1, \ldots, A_4, B_1, C_1$ are new symbols.

The productions in (1) shows how to generate a number of Z's bounded by the number of Y's after one X is converted to X'. The productions in (2) change A_4 to Y, whereas the productions in (3) change A_4 to Y'. The productions in (4) is used to convert remaining X's and Y's to X' and Y'.

It is not hard to see that $\$X^eY^f \Rightarrow_P^* \text{¢}w$ with $w \in \{Z, X', Y'\}^\oplus$ iff $w = \text{¢}Z^iX'^eY'^f$, where $i, e, f \in \mathbb{N}_0$ and $i \leqslant ef$. ▌

In view of Lemma 1.2 we see that if we want to weakly compute U_i^s, then we need several copies of the argument. We avoid this since according to Definition 0.2 we cannot erase symbols by c.s. com. productions.

Lemma 1.3. *Let X be some symbol and* \$, ¢ *be two special symbols. Further let Z, X' be some other symbols. Then there is a set P of c.s. com. productions such that $\$X^e \Rightarrow_P^* \text{¢}w$ with $w \in \{Z, Z'\}^\oplus$ iff $w = Z^iX'^e$, where $i, e \in \mathbb{N}_0$ and $i \leqslant e^s$ (s is some fixed integer $\geqslant 1$.)*

Proof. Consider the case $s = 2$. Define P to be the following set of productions:

(1) $\$ \to A_1,\ A_1X \to A_1ZX' \mid A_1X',\ A_1 \to \text{¢},$

(2) $\$ \to A_2,\ A_2X \to A_3X_2X_3,\ A_3X \to A_3X_1X_2,\ A_3 \to C,$

(3) Productions for $CX_1^{e-1}X_2^e \Rightarrow^* DZ^iX_1'^{e-1}X_2'^e, i \leqslant e(e-1)$.

(4) $D \to \text{¢},\ X_1'\text{¢} \to Z\text{¢},\ X_3\text{¢} \to Z\text{¢},\ X_2'\text{¢} \to X'\text{¢}.$

From Lemma 1.2 it follows that productions for (3) can be constructed. Now, (1) implies that $\$X^e \Rightarrow_P^* \text{¢}Z^iX'^e$, $i \leqslant e$. Productions in (2) generate two copies for the arguments for the multiplication performed by (3). Thus with (2), (3), and (4) we have that $\$X^e \Rightarrow_P^* \text{¢}w$ with $w \in \{Z, X'\}^{\oplus}$ iff $w = \text{¢}Z^iX'^e$, where $e \leqslant i \leqslant e^2$. Thus, P weakly computes the monomial U^2. A generalization for arbitrary s is straightforward. ▌

The next lemma shows how we can weakly compute the sum of two integers by c.s. com. productions.

LEMMA 1.4. *Let X, Y, Z be some symbols and \$, ¢ be two special symbols. Further let X', Y' be some other symbols. Then there is a set P of c.s. com. productions such that $\$X^eY^f \Rightarrow_P^* \text{¢}w$ with $w \in \{Z, X', Y'\}^{\oplus}$ iff $w = Z^iX'^eY'^f$, where i, $e, f \in \mathbb{N}_0$ and $i \leqslant e+f$.*

Proof. Define P to be the following set of productions

$$\$ \to B, \qquad XB \to X'A, \qquad A \to ZB, \qquad YB \to Y'A, \qquad B \to \text{¢}.$$

Clearly, these are c.s. com. productions and it holds that $\$X^eY^f \Rightarrow_P^* \text{¢}w$ with $w \in \{Z, X', Y'\}^{\oplus}$ iff $w \in Z^iX'^eY'^f$, where i, $e, f \in \mathbb{N}_0$, and $i \leqslant e+f$. ▌

We now show how monomials can be weakly computed by c.s. com. productions.

LEMMA 1.5. *Let $M(U_1,..., U_n)$ be a monomial. Let $X_1,..., X_n$ be some symbols and \$, ¢ be two special symbols. Further let Z, $X_1',..., X_n'$ be some other symbols. Then there is a set P of c.s. com. productions s.t. $\$X_1^{e_1} \cdots X_n^{e_n} \Rightarrow_P^* \text{¢}w$ iff $w = \text{¢}Z^iX_1'^{e_1} \cdots X_n'^{e_n}$, where i, $e_1,..., e_n \in \mathbb{N}_0$ and $i \leqslant M(e_1,..., e_n)$.*

Proof. We apply Lemmas 1.2 and 1.3. By Lemma 1.3 we see that $U_i^{s_i}$ can be weakly computed by c.s. com. productions. To weakly compute $M(U_1,..., U_n) = U_1^{s_1} \cdots U_n^{s_n}$ we apply Lemma 1.2, where the inputs are the outputs by the weak computation of $U_1^{s_1},..., U_n^{s_n}$.

Consider, for simplicity, $U_1^{s_1}U_2^{s_2}$, $s_1, s_2 \geqslant 1$. By Lemma 1.3 there are two sets, say P_1, P_2, of c.s. com. productions such that

$$\$'X_1^e \overset{*}{\underset{P_1}{\Rightarrow}} \text{¢}'Z_1^iY_1^e$$

and

$$\text{¢}'X_2^f \overset{*}{\underset{P_2}{\Rightarrow}} @Z_2^j Y_2^f,$$

where $i \leqslant e^{s_1}$ and $j \leqslant f^{s_2}$ and the symbols are appropriately chosen. We consider two cases.

Case 1. Either $e \leqslant 1$ or $f \leqslant 1$. W.l.o.g. let $e \leqslant 1$. We can construct a set P_0 of c.s. com. productions that perform the following task: (i) if $e=0$ then $\$X_2^f \Rightarrow^* \text{¢}X_2'^f$, (ii) if $e=1$ then $\$X_2^f \Rightarrow^* \text{¢}Z^jX_2'^f$, where $j \leqslant f^{s_2}$ and $f \geqslant 1$. Other subcases can be treated in a similar way.

Case 2. Both e and f are $\geqslant 2$. This can be tested by the production $\$X_1^2X_2^2 \rightarrow \$'X_1^2X_2^2$. We want to obtain

$$@Z_1^iZ_2^j \overset{*}{\underset{P_3}{\Rightarrow}} \text{¢}Z^k$$

such that $k \leqslant ij$ for some set P_3 of c.s. com productions.

Observe that using the techniques in previous lemmas we can construct a set P_3 of c.s. com. productions that perform the following task:

(1) either $@Z_1^iZ_2^j \Rightarrow^* \text{¢}Z^{i+j}$,

(2) or if $i \geqslant 3$ and $j \geqslant 3$, then perform

$$@Z_1^iZ_2^j \overset{*}{\Rightarrow} @'Z_1'^{i-2}Z_2'^{j-1}A_1^2A_2^1;$$

$$@'Z_1'^{i-2}Z_2'^{j-1} \overset{*}{\Rightarrow} \#Z^lB_1^{i-2}B_2^{j-1}, \qquad \text{where} \qquad l \leqslant (i-2)(j-1);$$

$$\#B_2^{j-1} \overset{*}{\Rightarrow} \#'C_2^{(j-1)+m}, \qquad \text{where} \qquad m \leqslant j-2;$$

$$\#'B_1^{i-2}C_2^{(j-1)+m}A_1^2A_2^1 \overset{*}{\Rightarrow} \text{¢}Z^{i+j+m}.$$

Note that by (2) we have $@Z_1^iZ_2^j \Rightarrow^* \text{¢}Z^k$, where $i+j \leqslant k \leqslant ij$. On the other hand, since i, j can be any value $\leqslant e^{s_1}$ and $\leqslant f^{s_2}$, respectively, it follows that by P_1, P_2, P_3 in Case 2 and P_0 in Case 1, $U_1^{s_1}U_2^{s_2}$ can be weakly computed.

The above construction can be generalized for arbitrary monomial. This completes the proof of Lemma 1.5. ▌

COROLLARY 1.6. *Lemma* 1.5 *holds if* $M(U_1,..., U_n)$ *is replaced by* $cM(U_1,..., U_n)$, *where* $c \geqslant 1$ *is any integer.*

Proof. Straightforward. ▌

We are now able to show that polynomials in $\mathbb{N}_0[U_1,..., U_n]$ can be weakly computed by c.s. com. grammars.

PROPOSITION 1.7. *Let $Q(U_1,..., U_n)$ be a polynomial in $\mathbb{N}_0[U_1,..., U_n]$. Then there is a c.s. com. grammar G^c with terminal alphabet $T = \{¢, t_1,..., t_n, t\}$ such that*

$$¢t_1^{a_1} \cdots t_n^{a_n} t^b \in L(G^c) \qquad \textit{iff} \qquad (a_1,..., a_n, b) \in G(Q).$$

Proof. From Corollary 1.6 and Lemma 1.4, G^c can be easily constructed. The details are left to the reader. ▌

THEOREM 1.8. *INEQ-CSCG is undecidable.*

Proof. From Lemma 1.1 and Proposition 1.7 it follows that the problem of determining whether $L(G_1^c) \subseteq L(G_2^c)$, where G_1^c, G_2^c are c.s. com. grammars, is undecidable, since PGIP is recursively reducible to it. Observe that

$$L(G_1^c) \subseteq L(G_2^c) \qquad \text{iff} \qquad L(G_2^c) = L(G_1^c) \cup L(G_2^c).$$

Thus, INEQ-CSCG is undecidable. ▌

2. THE INEQUIVALENCE PROBLEMS FOR CONTEXT-FREE COMMUTATIVE GRAMMARS

In this section we study the complexity of various versions of the inequivalence problems for context-free (c.f.) commutative grammars. The main result is that the general inequivalence problem for this grammar class is decidable in nondeterministic exponential time.

Let INEQ-CFCG (FINEQ-CFCG) denote the (finite) inequivalence problem for c.f. com. grammars. From the result that the uniform word problem for c.f. com. grammars (UWP-CFCG) is *NP*-complete we will see later that it is not hard to show that INEQ-CFCG is in the second level (Σ_2^e) of the exponential time hierarchy. To show that INEQ-CFCG is in nondeterministic exponential time (Σ_1^e) we need some technical results concerning commutative images of c.f. languages, which are semilinear sets by Parikh theorem (cf. [2, p. 146]).

In the following let $G^c = (N, T, S, P^c)$ be a c.f. com. grammar. Further let G be a reduced c.f. grammar which induces G^c (cf. [6, Sect. 1]), i.e., $\psi(L(G)) = L(G^c)$, where ψ is the Parikh mapping from T^* onto $T^\oplus$, the free com. monoid generated by T.

Construction of a Representation for $\psi(L(G))$

Following [1] we define semilinear sets in $T^{\oplus}$ as follows:

For a finite subset $W \subseteq T^{\oplus}$ let $W^{\oplus}$ denote the submonoid generated by W in $T^{\oplus}$. A subset $L \subseteq T^{\oplus}$ is called linear (lin.) if $L = c + W^{\oplus}$ for some $c \in T^{\oplus}$ and some finite subset $W \subseteq T^{\oplus}$. A subset $SL \subseteq T^{\oplus}$ is semilinear (s.l. for short) if it is a finite union of lin. sets.

If $SL = (c_1 + W_1^{\oplus}) \cup \cdots \cup (c_m + W_m^{\oplus})$, then $\{(c_1, W_1),..., (c_m, W_m)\}$ is called a representation of SL; c_i's (W_i's) are constants (period systems).

A representation of $L(G^c) = \psi(L(G))$ as a s.l. set may be obtained from the proof of Parikh theorem as follows (cf. [2, p. 146]):

Let V be any subset of $N \cup T$ containing T and S. Let s be card(V). Consider the set L_V of all words w in $L(G)$ such that in some generation tree of $S \Rightarrow_G^* w$ the nonterminals which are node labels are exactly $V \backslash T$. Since $L(G)$ is the union of all such L_V and there are only exponentially many such L_V, in order to obtain a representation for $\psi(L(G))$, it suffices to show how to obtain a representation for $\psi(L_V)$.

Let V be fixed and consider some nonterminal $X \in V$. Define two sets E and D_X as follows: A word w is in D_X iff the following two conditions are satisfied:

(1) w contains exactly one occurrence of X and no other nonterminals.

(2) There is a generation tree of $X \Rightarrow_G^* w$ whose node labels belong to V such that none of them occurs more than $s+2$ times in any path of the tree.

A word $w \in T^*$ is in E iff the following condition holds:

There is a derivation tree of $S \Rightarrow_G^* w$ whose interior node labels are exactly $V \backslash T$ and none of them occurs more than $s+2$ times in any path of the tree.

Let G denote $\psi(E)$ and H denote $\psi(\bigcup_X D_X)$. (Note that for $w \in D_X$, $\psi(w)$ is defined to be $\psi(w')$, where w' is obtained from w by erasing X in w.) Then we have

$$\psi(L_V) = G + H^{\oplus}. \qquad (*)$$

Taking the union of sets of the form $G + H^{\oplus}$, where the union runs over all $V \subseteq N \cup T$ satisfying the above requirement, we obtain a s.l. set representation for $\psi(L(G)) = L(G^c)$.

Complexity of FINEQ-CFCG

From the above construction we can easily show that FINEQ-CFCG is Σ_2^p-complete.

PROPOSITION 2.1. *FINEQ-CFCG is in Σ_2^p.*

Proof. Let G^c be a c.f. com. grammar. If $L(G^c)$ is finite, then in the s.l. set representation of $L(G^c)$ constructed above, all period systems are empty. The constants, encoded as in [6, *Sect.* 1], have sizes polynomially bounded in terms of $\|G^c\|$, the size of G^c, since derivation trees for words in E have depths that are linearly bounded in the number of variables of G^c. Therefore the following fact holds.

Fact. Let G_1^c and G_2^c be two c.f. com. grammars generating finite languages. Then $L(G_1^c) \neq L(G_2^c)$ iff there is some com. word w in $\Delta(G_1^c, G_2^c) := [L(G_1^c) \backslash L(G_2^c)] \cup [L(G_2^c) \backslash L(G_1^c)]$ such that

$$\|w\| \leqslant Q(\|G_1^c\| + \|G_2\|),$$

where Q is a fixed polynomial. ▌

Since UWP-CFCG (the uniform word problem) is in NP, it is straightforward to see that FINEQ-CFCG is in Σ_2^p:

— Guess a polynomially bounded com. word w.

— Verify that $w \in \Delta(G_1^c, G_2^c)$.

This completes the proof of Proposition 2.1. ▌

To show that FINEQ-CFCG is Σ_2^p-hard we reduce the inequivalence problem for integer expressions, denoted by INEQ-N, to FINEQ-CFCG. Since INEQ-N is known to be Σ_2^p-complete (cf. [12]), it will follow that FINEQ-CFCG is Σ_2^p-hard.

Integer expressions are well-formed parenthesized expressions involving nonnegative integer constants written in binary notation and two binary operations: addition $(+)$ and union $(\cup)$. Integer expressions defined subsets of nonnegative integers recursively as follows: $L(a) = \{a\}$, $a \in \mathbb{N}_0$; $L((E_1 \cup E)) = L(E_1) \cup L(E_2)$ and $L((E_1 + E_2)) = L(E_1) + L(E_2) = \{x + y \mid x \in L(E_1) \text{ and } y \in L(E_2)\}$.

PROPOSITION 2.2. *FINEQ-CFCG is Σ_2^p-hard.*

Proof. Since derivations in a c.f. com. grammar with a single-letter terminal alphabet can simulate (binary) addition and union, the log-space reduction of INEQ-N to FINEQ-CFCG is straightforward. Thus FINEQ-CFCG is Σ_2^p-hard. ▌

COROLLARY 2.3. *INEQ-CFCG is Σ_2^p-hard.*

Proof. Obvious. ▌

From Propositions 2.1 and 2.2 we obtain

THEOREM 2.4. *FINEQ-CFCG is Σ_2^p-complete.*

Remark. Note that the finite inequivalence problem for (noncommutative) c.f. grammars is complete for nondeterministic exponential time[3] (NEXPTIME) under log-space reduction, since it is obviously in NEXPTIME and it is NEXPTIME-hard by Theorem 4.5 in [5].

Before showing that INEQ-CFCG is in NEXPTIME we make some remarks. In Lemma 2.9 below, we will show that two c.f. com. grammars are inequivalent iff there is an exponentially bounded com. word witnessing that inequivalence. Hence, it follows that INEQ-CFCG is in Σ_2^e, the second level of the exponential hierarchy, since UWP-CFCG is in *NP*. To get the NEXPTIME upper bound we need to show that if the witnessing word is exponentially bounded, then the test for membership can be done deterministically. We need some technical results about the s.l. set representation of $L(G^c)$.

Some Observations about the S.L. Set Representation of $L(G^c)$

In the following we provide some technical results for proving the NEXPTIME upper bound of INEQ-CFCG. Observe that the results derived in [6] cannot be applied to obtain a deterministic test for membership, since a com. derivation word which is also of exponentially bounded size must be guessed in the test for membership. (This implies a Σ_2^e upper bound.)

LEMMA 2.5. *In the representation of $L(G^c)$ as a s.l. set the constants and periods have sizes polynomially bounded in terms of $\|G^c\|$.*

Proof. Consider V and L_V in the construction above. Since the trees generating words in E and D_X, $X \in V$, have depths bounded by $(s+2)^2$, $s = \text{Card}(V) \leqslant \text{Card}(N \cup T)$, the lemma follows. ▌

From the construction above let us consider $V \subseteq N \cup T$, L_V, $\psi(L_V)$, where $G = (N, T, S, P)$ is the c.f. grammar inducing $G^c = (N, T, S, P^c)$. We now show how to obtain from the representation (*) $\psi(L_V) = G + H^{\oplus}$ another representation of the form

$$\psi(L_V) = \bigcup c + W^{\oplus}$$

where the union runs over $c \in \psi(L_V)$ with polynomially bounded size and subsets $W \subseteq H$ with $\leqslant k$ linearly independent periods, where $k := \text{Card}(T)$.

[3] NEXPTIME $= \bigcup_{k \geqslant 1} \text{NTIME}(c^{n^k})$.

PROPOSITION 2.6. *Let $G=(N, T, S, P)$ be a c.f. grammar inducing G^c. Let $V \subseteq N \cup T$ such that V contains $\{S\}$ and T. Then*

$$\psi(L_V)=\bigcup c+W^{\oplus}$$

where the union runs over $c \in \psi(L_V)$ with polynomially bounded size (in terms of $\|G^c\|$) and subsets $W \subseteq H$ $(=\psi(\bigcup_{X \in V} D_X))$ with $\leqslant k(=\operatorname{Card}(T))$ linearly independent polynomially bounded periods.

In the following, we proceed to show Proposition 2.6. This will be done via Lemmas 2.7 and 2.8. Obviously, we only need to show that $g+H^{\oplus}$, $g \in G$, has a representation of the form

$$g+H^{\oplus}=\bigcup c+W^{\oplus}$$

where the union runs over $c \in g+H^{\oplus}$ with polynomially bounded size and subsets $W \subseteq H$ with $\leqslant k$ linearly independent periods.

We need some technical notations. In the following two lemmas we may consider $T^{\oplus}$ as $\mathbb{N}_0^k$, since they are isomorphic. We also regard $\mathbb{N}_0^k$ as subset of $\mathbb{Q}^k$, where $\mathbb{Q}$ is the set of rationals. Further $\mathbb{Q}_+$ denotes the nonnegative rationals. $C(H)$ denotes the cone generated by H:

$$C(H):=\left\{\sum_{h \in H} r_h h \,\middle|\, r_h \in \mathbb{Q}_+\right\}.$$

Let $M(g, H)$ denote the set of lattice points in $g+C(H)$, i.e., $M(g, H)=(g+C(H)) \cap \mathbb{N}_0^k$.

LEMMA 2.7. *$M(g, H)$ is a s.l. set with representation of the form*

$$M(g, H)=\bigcup F+W^{\oplus},$$

where the union runs over subsets $W \subseteq H$ of $\leqslant k$ linearly independent periods and each element $f \in F$ has the form

$$f=g+\sum_{i=1}^{n} r_i h_i, \qquad 0 \leqslant r_i<1,$$

with linearly independent vectors $h_1, \ldots, h_n \in H$.

Proof. Let $x \in M(g, H)$. Since $x \in g+C(H)$, by Caratheodory's theorem for cones (cf. [13, p. 35]), there are n $(\leqslant k)$ linearly independent vectors $h_1, \ldots, h_n$ in H such that

$$x=g+\sum_{i=1}^{n} \rho_i h_i, \qquad \rho_i \in \mathbb{Q}_+.$$

This may be written as

$$x = g + \sum_{i=1}^{n} (\rho_i - \lfloor \rho_i \rfloor) h_i + \sum_{i=1}^{n} \lfloor \rho_i \rfloor h_i.$$

Letting f be $g + \sum_{i=1}^{n} (\rho_i - \lfloor \rho_i \rfloor) h_i$, Lemma 2.7 follows. ▌

LEMMA 2.8. *$g + H^{\oplus}$ has a representation of the form*

$$\bigcup c + W^{\oplus}$$

where the union runs over $c \in g + H^{\oplus}$ with polynomially bounded sizes (in terms of $\|G^c\|$) and subsets $W \subseteq H$ with $\leqslant k$ linearly independent vectors.

Proof. Consider the intersection

$$M(g, H) \cap (g + H^{\oplus}) = (\bigcup F + W^{\oplus}) \cap (g + H^{\oplus})$$

which is exactly $(g + H^{\oplus})$. It suffices to obtain a representation for $(f + W^{\oplus}) \cap (g + H^{\oplus})$ with the desired property.

Consider the minimal nonnegative integer solutions set of the system of equations with integer coefficients

$$g^{\mathrm{T}} + Hx^{\mathrm{T}} = f^{\mathrm{T}} + Wy^{\mathrm{T}},$$

where the vectors in H and W are written as column vectors and x^{T}, y^{T} are appropriate vectors of unknowns. Let Sol denote this set. Define

$$C = \{f + Wy \mid \exists x\colon (x, y) \in \mathrm{Sol}\}.$$

Then from Theorem 5.6.1, [2, p. 180], we have

$$(f + W^{\oplus}) \cap (g + H^{\oplus}) = C + W^{\oplus}.$$

We now show that $c \in C$ has polynomially bounded size (in terms of $\|G^c\|$). The above equation system has rank $l \leqslant k$. Let m be the number of unknowns. Then the coefficients of vectors in Sol is bounded by $(m+1) \times N$, where N is the maximum of the absolute values of all $l \times l$ subdeterminants of the system. (This is proved in [10] by Gathen and Sieveking.) A simple calculation shows that every $c \in C$ has size polynomially bounded in terms of $\|G^c\|$. This completes the proof of Lemma 2.8. ▌

From Lemma 2.8, Proposition 2.6 follows.

INEQ-CFCG is in NEXPTIME

We are now able to show that INEQ-CFCG is in NEXPTIME. We need a lemma.

LEMMA 2.9. *Let G_1^c and G_2^c be two c.f. com. grammars. Then $L(G_1^c) \neq L(G_2^c)$ iff there is a com. word $w \in \Delta(G_1^c, G_2^c)$ such that*

$$\|w\| \leqslant 2^{Q(\|G_1^c\| + \|G_2^c\|)}$$

for some fixed polynomial Q.

Proof. We outline a proof sketch and omit the details. The idea of the proof is as follows. We apply Parikh theorem and the result about s.l. sets obtained in [9] (or [8]). First consider the s.l. set representations of $L(G_1^c)$ and $L(G_2^c)$ obtained by the construction at the beginning: From Lemma 2.5 it follows that these representations have sizes that are exponentially bounded in terms of $\|G_1^c\|$ and $\|G_2^c\|$, respectively. Let RP_1 and RP_2 denote these representations.

Now, applying the main lemma in [9] we have that the symmetric difference $\Delta(G_1^c, G_2^c)$ is not empty iff there is some com. word $w \in \Delta(G_1^c, G_2^c)$ such that

$$\|w\| \leqslant Q_1(\text{size}(RP_1) + \text{size}(RP_2)).$$

Thus in terms of $\|G_1^c\| + \|G_2^c\|$ the size of w is exponentially bounded, and Lemma 2.9 follows. ▌

THEOREM 2.10. *INEQ-CFCG is in NEXPTIME.*

Proof. We apply Proposition 2.6 and Lemma 2.9. Consider the following nondeterministic algorithm. Let G_1^c and G_2^c be two c.f. com. grammars in the input:

— Guess an exponentially bounded com. word w.

— Check that $w \in \Delta(G_1^c, G_2^c)$.

Applying Proposition 2.6 we now show how the test $w \in \Delta(G_1^c, G_2^c)$ can be carried out deterministically. We need only to show this for the test $w \in L(G_1^c)$. Let $k = \text{Card}(T)$. Consider the following algorithm:

```
*for* all $V \subseteq N \cup T$ such that $S \in V$ and $T \subseteq V$ *do*
  *for* all com. word $c \in T^{\oplus}$ which is polynomially bounded in terms
     of $\|G^c\|$ *do*
     *begin* check that $c \in \psi(L_V)$;
           *for* all subsets $W = \{h_1, ..., h_n\}$, $n \leqslant k$, of linearly
              independent periods from $\psi(\bigcup_{X \in V} D_X)$ *do*
              *if* $w = c + \sum_{i=1}^{n} \lambda_i h_i$, $\lambda_i \in \mathbb{N}_0$ *then* accept
     *end.*
```

We have to show that if w is exponentially bounded in terms of $\|G_1^c\|$, then the above algorithm is in deterministic exponential time (in terms of $\|G_1^c\|$).

Since $\|c\|$ is polynomially bounded in terms of $\|G_1^c\|$, the test $c \in \psi(L_V)$ can be done nondeterministically in time polynomial in $\|G_1^c\| + \|c\|$ (and hence in $\|G_1^c\|$), in view of the fact that UWP-CFCG is in *NP*. Thus this test can be done deterministically in exponential time.

The innermost for-loop can be carried out in deterministic exponential time, since there are at most an exponential number of subsets of $\leqslant k$ periods and since verifying $w = c + \sum_{i=1}^{n} \lambda_i h_i$ can be done deterministically in time polynomial in $\|w\| + \|h_1\| + \cdots + \|h_n\|$, and hence exponential in $\|G_1^c\|$.

Thus we conclude that $w \in \Delta(G_1^c, G_2^c)$ can be done deterministically in exponential time. Hence INEQ-CFCG $\in$ NEXPTIME. This completes the proof of Theorem 2.10. ∎

Remark. Unfortunately, there is still a gap between the upper and lower bounds for INEQ-CFCG. It seems that Lemma 2.9 can be strengthened so that the witnessing com. word has a polynomially bounded size. (In that case, such a proof would require a different technique.) This would imply that INEQ-CFCG is in Σ_2^p and hence Σ_2^p-complete. Such a result is interesting, since a Σ_2^p upper bound for INEQ-CFCG also provides the Σ_2^p upper bounds for the inequivalence problems for s.l. sets and context-free grammars with 1-letter terminal alphabet (the reduction of s.l. set inequivalence to c.f. com. grammar inequivalence is straightforward and the other reduction is trivial), whose proofs employ completely different techniques and are non-trivial (cf. [7, 9]).

Open Problem. Is INEQ-CFCG in Σ_2^p?

Remark. Notice that INEQ-CFCG is closely related to the commutative inequivalence of (noncommutative) c.f. grammars. From the results in [7] it is known that the latter is Σ_2^p-hard, whereas completeness is (to the author's knowledge) an open question. In the following we will see that a Σ_2^p upper bound for one problem implies such an upper bound for the other.

The Case of Single Letter Terminal Alphabet CFCGs

Let INEQ-CFCG-$\{0\}$ (FINEQ-CFCG-$\{0\}$) denote INEQ-CFCG (FINEQ-CFCG) with the restriction that the terminal alphabets are singleton sets.

Theorem 2.11. *FINEQ-CFCG-$\{0\}$ is Σ_2^p-complete.*

Proof. Notice that the reduction in Proposition 2.2 outputs grammars with single-letter terminal alphabets. ∎

THEOREM 2.12. *INEQ-CFCG-$\{0\}$ is Σ_2^p-complete.*

Proof. We only need to show that INEQ-CFCG-$\{0\}$ is in Σ_2^p. To this end we apply the result in [7] that deciding inequivalence of c.f. grammars with 1-letter terminal alphabet is Σ_2^p-complete. We have to show that a c.f. com. production can be simulated by only a polynomial number of c.f. productions. This can be accomplished as follows. Consider a com. production of the form $X \to Y_1^{2^{e_1}} \cdots Y_m^{2^{e_m}}$, where $e_1, ..., e_m \in \mathbb{N}_0$. Define the following c.f. productions:

```
$X \to A_{1,0} \cdots A_{m,0}$
for each $j = 1, ..., m$ do
  begin
    for each $l = 0, ..., e_j - 1$ do
      define $A_{j,l} \to A_{j,l+1} A_{j,l+1}$;
    define $A_{j,e_j} \to Y_j$
  end
```

Obviously this set of c.f. productions simulates the com. production above. From this observation it can easily be seen that INEQ-CFCG-$\{0\}$ is log-space reducible to the inequivalence problem for c.f. grammars with 1-letter terminal alphabet. Since the latter is in Σ_2^p, it follows that INEQ-CFCG-$\{0\}$ is in Σ_2^p, too. Hence it is Σ_2^p-complete. ∎

3. THE INEQUIVALENCE PROBLEMS FOR REGULAR COMMUTATIVE GRAMMARS AND RATIONAL EXPRESSIONS IN COMMUTATIVE MONOIDS

In this section we classify the complexity of the inequivalence problems for regular (reg.) com. grammars (INEQ-RCG) and rational expressions in com. monoids (cf. [6] for definitions). The latter problem is denoted by INEQ-RE(V, k) if the com. monoid is finitely generated by V and k is the upper bound for the star heights of the expressions, and by INEQ-RE(V) if there is no restriction on the star heights.

THEOREM 3.1. *For any fixed finite alphabet V and nonnegative integer k, INEQ-RE(V, k) is Σ_2^p-complete.*

Proof. We first show that this problem is Σ_2^p-hard. To this end consider the restricted case that $V = \{0\}$ is a singleton set and $k = 0$. Obviously, rational expressions in $\{0\}^{\oplus}$ without stars can simulate integer expressions. Thus, INEQ-RE$(\{0\}, 0)$ is Σ_2^p-hard.

We now show that INEQ-RE(V, k) is in Σ_2^p for any fixed k and finite alphabet V. The idea is to show that rational expressions with bounded star heights have s.l. set representations with polynomially bounded sizes.

This can be done recursively as follows. Let α be a rational expression in $V^{\oplus}$. Let $\mathrm{RP}(\alpha)$ be the s.l. set representation for α which is to be constructed:

(1) If $\alpha = w$, $w \in V^{\oplus}$, then $\mathrm{RP}(\alpha) = \{(w, \varnothing)\}$

(2) If $\alpha = (\alpha_1 \cup \alpha_2)$, then $\mathrm{RP}(\alpha) = \mathrm{RP}(\alpha_1) \cup \mathrm{RP}(\alpha_2)$

(3) $\alpha = (\alpha_1 \cdot \alpha_2)$. Let $\mathrm{RP}(\alpha_1) = \{(c_1, W_1),..., (c_m, W_m)\}$ and

$$\mathrm{RP}(\alpha_2) = \{(d_1, U_1),... (d_n, U_n)\}.$$

Since α defines the set

$$\bigcup_{i,j} (c_i + W_i^{\oplus}) + (d_j + U_j^{\oplus})$$

it suffices to show how to obtain a representation for

$$(c_i + W_i^{\oplus}) + (d_j + U_j^{\oplus}).$$

This can be represented by

$$(c_i + d_j) + (W_i \cup U_j)^{\oplus}.$$

(4) $\alpha = (\alpha_1)^*$, where $\mathrm{RP}(\alpha_1) = \{(c_1, W_1),..., (c_m, W_m)\}$.
Since α defines the set

$$[(c_1 + W_1^{\oplus}) \cup \cdots \cup (c_m + W_m^{\oplus})]^{\oplus}$$

which is

$$(c_1 + W_1^{\oplus})^{\oplus} + \cdots + (c_m + W_m^{\oplus})^{\oplus},$$

we only need, by induction and (3), to consider the case $m = 1$. Obviously, $(c_1 + W_1^{\oplus})^{\oplus}$ defines the set

$$\{\varepsilon\} \cup [c_1 + (\{c_1\} \cup W_1)^{\oplus}].$$

Now, if k is fixed, then every rational expression α has a s.l. set representation $\mathrm{RP}(\alpha)$ whose size is polynomially bounded in $\|\alpha\|$. (Notice that this fact, according to our argument, does not hold if k is not fixed.) Further $\mathrm{RP}(\alpha)$ can be computed deterministically in polynomial time. Thus INEQ-RE(V, k) is in Σ_2^p, since the inequivalence problem for s.l. sets is in Σ_2^p (cf. [8, and 9]). This completes the proof of Theorem 3.1. ▌

Corollary 3.2. *INEQ-RE*($\{0\}$, k) *is Σ_2^p-complete for any fixed k.*

Proof. Follows from the above proof. ▌

Since INEQ-RCG and INEQ-RE are polynomially related, we obtain from the results in previous section.

THEOREM 3.3. (1) *INEQ-RE(V) and INEQ-RCG are in NEXPTIME,*

(2) INEQ-RE(*V*) *and INEQ-RCG are Σ_2^p-hard,*

(3) INEQ-RE({0}) *and* INEQ-RCG-{0} *is Σ_2^p-complete,*

(4) FINEQ-RE({0}) *and* FINEQ-RCG-{0} *is Σ_2^p-complete,*

where (4) *is the finite version of* (3).

4. CONCLUDING REMARKS

In this paper we have investigated the complexity of the equivalence problems for various classes of commutative grammars. The results are summarized in Table I. For type 0 com. grammars the inequivalence problem is recursively enumerable (r.e.), since it is now known that the uniform word problem is decidable, as shown recently by Mayr (cf. [6]). From the results of Van Leeuwen [11] and Hopcroft and Pansiot [4] it follows that this inequivalence problem for type 0 com. grammars is decidable if the number of symbols is bounded by 5. It would be interesting to extend this bound. Specifically, we do not know whether this problem is still decidable when the number of symbols is bounded.

TABLE I

Inequivalence problem		Upper bound	Lower bound
Type 0 and c.s. com. grammars		r.e.	undecidable
Context-free and regular commutative grammars	General	NEXPTIME	Σ_2^p-hard
	Finite	Σ_2^p	Complete
	One terminal letter	Σ_2^p	Complete
Rational expressions	General	NEXPTIME	Σ_2^p-hard
	Bounded star heights	Σ_2^p	Complete
	Finite	Σ_2^p	Complete
	One terminal letter	Σ_2^p	Complete

Acknowledgments

The author wishes to thank a referee for helpful suggestions that improve the presentation of this paper.

Received April 5, 1984; accepted May 22, 1985

References

1. Eilenberg, S. and Schützenberger, 1969), Rational sets in commutative monoids, *J. Algebra* **13**, 173–191.
2. Ginsburg, S. (1966), "The Mathematical Theory of Context-Free Languages," McGraw–Hill, New York.
3. Hack, M. (1976), The equality problem for vector addition systems is undecidable, *Theoret. Comput. Sci.* **2**, 77–95.
4. Hopcroft, J. and Pansiot, J. (1979), On the reachability problem for 5-dimensional vector addition systems, *Theoret. Comput. Sci.* **8**, 135–159.
5. Hunt, H., Rosenkrantz, D. and Szymanski, T. (1976), On the equivalence, containment and covering problems for the regular and context-free languages, *J. Comput. System Sci.* **12**, 222–268.
6. Huynh, D. (1983), Commutative grammars: The complexity of uniform word problems, *Inform. Contr.* **57**, 21–39.
7. Huynh, D. (1984), Deciding the inequivalence of context-free grammars with 1-letter terminal alphabet is Σ_2^p-complete, *Theoret. Comput. Sci.* **33**, 305–326.
8. Huynh, D. (1982), The complexity of semilinear sets, *Elektron. Informationsverarb. Kybernet.* **18**, 291–338.
9. Huynh, D. (in press), A simple proof for the Σ_2^p upper bound of the inequivalence problem for semilinear sets, *Elektron. Informationsverarb. Kybernet.*
10. von zur Gathen, J. and Sieveking, M. (1978), A bound on solutions of linear integer equalities and inequalities, *Proc. Amer. Math. Soc.* **72**, 155–158.
11. van Leeuwen, J. (1974), A partial solution to the reachability problem for vector addition systems *in*, "Proc. 6th ACM Sympos. Theory of Computing," pp. 303–307.
12. Stockmeyer, L. and Meyer, A. (1973), Word problems requiring exponential time, *in* "Proc. 5th ACM Sympos. Theory of Computing," pp. 1–9.
13. Stoer, J. and Witzgall, C. (1970), "Convexity and Optimization in Finite Dimensions I," Springer-Verlag, Berlin/New York.

Printed in Belgium