INFORMATION AND CONTROL 57, 21–39 (1983)

Commutative Grammars: The Complexity of Uniform Word Problems

DUNG T. HUYNH*

Fachbereich Informatik, Universität des Saarlandes, Saarbrücken D-6600, West Germany

Commutative grammars are introduced, and various classes of commutative grammars are defined. The complexity of uniform word problems for commutative grammars is investigated.

0. INTRODUCTION

This paper is motivated by the work of Hotz (1980) on invariants of formal languages. Let $\mathscr{G}$ be a class of grammars and $\mathscr{M}$ be a set. A mapping I: $\mathscr{G} \to \mathscr{M}$ is called an invariant for $\mathscr{G}$ if I satisfies the following: $L(G_1) = L(G_2)$ implies $I(G_1) = I(G_2)$, where G_1 and G_2 are arbitrary grammars in $\mathscr{G}$. Consider now a simple invariant derived by Paritkh's theorem, which states that the commutative images of context-free languages are semilinear sets. That this is an invariant for context-free grammars can be easily seen. On the other hand, the proof of this theorem gives us an effective method for computing a representation of the semilinear set from the input grammar (cf. Hotz, 1980, for a more detailed discussion).

Another motivation of this work is the close connection between commutative grammars and Petri nets, which has been introduced as a mathematical model for parallel computations. Commutative grammars have been defined by Crespi-Reghizzi and Mandrioli (1976). Thus the classification of the complexity of commutative grammars also provides some complexity results for subclasses of Petri nets.

In defining commutative grammars we do not use the notions "bag" and "multiset" as in Crespi-Reghizzi and Mandrioli (1976). It seems simpler and more natural to work on free commutative monoids. The definitions are directly related to vector replacement systems, another formulation of Petri nets.

* Present address: Computer Science Department, Iowa State University, Ames, Iowa 50011.

0019-9958/83 $3.00

For a finite alphabet T let $T^{\oplus}$ denote the free commutative monoid generated by T. A subset L of $T^{\oplus}$ is called a commutative language. Commutative grammars are generating devices for commutative languages: we use free commutative monoid instead of free monoid.

We want to classify the complexity of the equivalence and the uniform word problems for commutative grammars. The complexity of the equivalence problems will be investigated in a forthcoming paper. The results of the present paper concern the classification of the complexity of the uniform word problems.

This paper consists of four sections. Commutative grammars are introduced in Section 1, in which various restricted classes of commutative grammars are also defined. Section 2 deals with the complexity of the uniform word problem for context-free commutative grammars. In Section 3 we derive complexity results for regular commutative grammars and rational expressions in free commutative monoids. Section 4 contains the complexity classification for context-sensitive commutative grammars. The last section summarizes the results of this paper and contains some concluding remarks.

1. Notations and Basic Definitions

In this section we introduce basic definitions and notations which will be used later. Let V be a finite alphabet. V^* denotes the free monoid generated by V, ε denotes the empty word, and $V^{\oplus} := V^* \backslash \{\varepsilon\}$. 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

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

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 $V^{\otimes}$ 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

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

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

In the following we introduce commutative grammars. Consider a phrase-structure grammar $G = (N, T, S, P)$, where

N is the set of nonterminals,

T is the set of terminals, $T \cap N = \varnothing$,

$S \in N$ is the axiom, and

$P \subset N^+ \times (N \cup T)^*$ is the finite set of productions.

Let $L(G)$ denote the language generated by G. Parallel to this definition we introduce commutative grammars and commutative languages.

DEFINITION 1.1. A 4-tuple $G^c = (N, T, S, P^c)$ is called a *commutative* (com.) *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.

Let $G^c = (N, T, S, P^c)$ be a com. grammar and $\alpha, \beta \in V^{\oplus}$, where $V = N \cup T$. α is said to *directly generate* β, written $\alpha \Rightarrow_{G^c} \beta$, iff there are $\alpha_1 \in V^{\oplus}$ and $p \in P^c$ such that $p = (\gamma, \delta)$ and $\alpha = \alpha_1 \gamma$, $\beta = \alpha_1 \delta$. $\Rightarrow^*_{G^c}$ denotes the reflexive and transitive closure of $\Rightarrow_{G^c}$. We also write $\Rightarrow$ and $\Rightarrow^*$ if G^c is understood.

If $\alpha \Rightarrow^* \beta$, $\alpha, \beta \in V^{\oplus}$, we say that α *generates* β. In this case there are some n and $\alpha_i \in V^{\oplus}$, $i = 1,..., n$ such that

$$D : \alpha = \alpha_0 \Rightarrow \alpha_1 \Rightarrow \cdots \Rightarrow \alpha_n = \beta.$$

D is called a *derivation* and n is the *length* of D. D is called a *terminal derivation* if $\alpha = S$ and $\beta \in T^{\oplus}$. Sentential forms are called *commutative sentential forms.*

Let $s := \#P^c$ be the number of productions in G^c ($\#M$ denotes the cardinality of the set M). Let $\Pi := \{\pi_1,..., \pi_s\}$ be a finite alphabet and ϕ: $\Pi \to P^c$ be a bijection. If $p = \phi(\pi_i)$ for some i, $1 \leqslant i \leqslant s$, then π_i is called the *name* of p. Consider some derivation

$$D: \qquad \alpha = \alpha_0 \Rightarrow \alpha_1 \Rightarrow \cdots \Rightarrow \alpha_n = \beta.$$

Let π_{i_j}, $j = 1,..., n$, be the name of the production applied at the jth derivation step in D. Then $\pi_D := \pi_{i_1} \cdots \pi_{i_n} \in \Pi^*$ is called the *derivation word* defined by D.

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

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

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

$$D(G^c) := \{\pi \in \Pi^*: \pi \text{ is some terminal word of } G^c\}.$$

According to the noncommutative case we shall consider classes of com. grammars.

DEFINITION 1.2. 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 of P^c.

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

(3) *context-free* (c.f.), 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.), 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 $L(G^c) = L$.

Let $G = (N, T, S, P)$ be a phrase-structure grammar. Consider the com. grammar induced by G, i.e., $G^c = (N, T, S, P^c)$, where

$$P^c = \{(\psi_V(\gamma), \psi_V(\delta)) : (\gamma, \delta) \in P\}.$$

Let $\psi_T(L(G))$ denote the com. image of $L(G)$ under the Paritkh mapping on T^*. In general $\psi_T(L(G)) = L(G^c)$ does not hold. (This can be easily demonstrated by a simple counterexample.) However, if G is a c.f. grammar, then this equality holds. Conversely, for every c.f. com. grammar $G^c = (N, T, S, P^c)$ there is a c.f. grammar $G = (N, T, S, P)$ such that G^c is the com. grammar induced by G and the equality $\psi_T(L(G)) = L(G^c)$ holds. Thus we have

PROPOSITION 1.3. *Let G be a c.f. grammar and G^c be the c.f. com. grammar induced by G. Then*

$$\psi_T(L(G)) = L(G^c)$$

holds, where T is the terminal alphabet of G and G^c.

[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.

For the proofs in the next section we shall make use of the notions "reducedness" and "ε-freedom" of c.f. com. grammars which are defined as in the noncommutative case. Let $G^c = (N, T, S, P^c)$ be a c.f. com. grammar. G^c is said to be *reduced* if for any $A \in N$ there are $w \in T^{\oplus}$ and $\alpha \in (N \cup T)^{\oplus}$ such that

$$S \overset{*}{\Longrightarrow} \alpha A \qquad \text{and} \qquad A \overset{*}{\Longrightarrow} w.$$

A production is said to be ε-production if its right-hand side in ε. G^c is said to be *ε-free*, if for any $A \in N$ the following holds:

$(A, \varepsilon) \in P^c$ iff $\varepsilon \in L(G^c)$, $A = S$ and S does not appear on the right-hand side of any production.

In Section 4 we shall consider the uniform word problem for rational expressions in free com. monoids which are defined as rational expressions in free monoids.

DEFINITION 1.4. Let V be a finite alphabet and M denote the monoid $V^{\oplus}$. Rational expressions in M are defined as follows.

(1) $\varnothing$ is a rational expression.

(2) For all $w \in M$, w is a rational expression.

(3) If E and F are rational expressions, then $(E \cup F)$ and $(E \cdot F)$ are rational expressions.

(4) If E is a rational expression, then (E^*) is a rational expression.

(5) Nothing else is a rational expression.

For a rational expression E in M, the language defined by E, which is denoted by $L(E)$, is defined inductively as follows. $L(w) := \{w\}$ for all $w \in M$, $L(\varnothing) := \varnothing$, $L(E \cup F) := L(E) \cup L(F)$, $L(E \cdot F) := L(E) \cdot L(F)$, and $L(E^*) := (L(E))^*$, where the star operation in free com. monoids is defined accordingly.

A subset of M which is defined by a rational expression is called a rational set. It is well known that in com. monoids the notion "rationality" and "semilinearity" are equivalent (cf. Eilenberg and Schützenberger, 1969).

DEFINITION 1.5. Let $\mathscr{G}$ be a class of com. grammars. The uniform word problem for $\mathscr{G}$ is the following decision problem: Given a rom. grammar $G^c = (N, T, S, P^c) \in \mathscr{G}$ and a com. word $w \in T^{\otimes}$ we need to determine whether $w \in L(G^c)$.

In the next sections we shall classify the complexity of the uniform word problem for c.s., c.f., reg. com. grammars. We shall also derive some results for the complexity of the uniform word problem for rational expressions in free com. monoids. For this aim it is necessary to define the size of the inputs.

DEFINITION 1.6. (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

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

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 $V := N \cup T$.

Concerning our problem the size of an input instance is the sum of the size of a com. word and the size of a com. grammar. (For complexity notions the reader is referred to some textbook.)

Remark 1.7. The uniform word problem for rational expressions in free com. monoids is defined in a similar way. In this case an input instance is a com. word and a rational expression. It is straightforward to define the size of a rational expression in a free com. monoid and hence the size of the inputs.

2. THE COMPLEXITY OF THE UNIFORM WORD PROBLEM FOR CONTEXT-FREE COMMUTATIVE GRAMMARS

In this section we classify the complexity of the uniform word problem for c.f. com. grammars. It will be shown that this problem is *NP*-complete. While it is straightforward to prove *NP*-hardness, the argument that it is in *NP* is more complex.

The basic proof idea consists of the following observations: Consider the c.f. com. grammar G^c and a com. word $\tilde{w}$. Let G be a c.f. grammar which induces G^c as its com. image. Then $\tilde{w} \in L(G^c)$ iff there is some w with $\psi(w) = \tilde{w}$ such that $w \in L(G)$. Now $w \in L(G)$ iff there is a terminal derivation tree in G with frontier w. Of course, such a derivation tree may be exponentially large, since in the input $\tilde{w}$ is written as a com. word encoded by its exponents in binary notation. The crucial point is that in the commutative case we may guess a "small" com. terminal derivation word in order to check whether $\tilde{w} \in L(G^c)$. There are some problems: (1) Which com. derivation words are com. *terminal* derivation words? (2) How to check that such a com. terminal derivation word generates $\tilde{w}$? and (3) Such a com. terminal derivation word must have polynomial size so that the guess can be performed in *NP*!

We will see that (2) is easy to solve. It can be reduced to the problem of checking whether a linear diophantine equation system has an integer solution which is known to be in nondeterministic polynomial time. Questions (1) and (2) will be discussed in subsections 2.1 and 2.2. Subsection 2.1 gives a characterization of com. terminal derivation words, and subsection 2.2 shows the *NP* upper bound.

2.1. *A Characterization of Commutative Terminal Derivation Words*

We first introduce some notations and technical definitions. In the following UWP-CSCG (UWP-CFCG, UWP-RCG) denotes the uniform word problem for c.s. com. (c.f. com., reg. com.) grammars.

This section is devoted to c.f. com. grammars, and we assume w.l.o.g. that all c.f. and c.f. com. grammars are reduced. (We will see in the proofs below that this is no restriction, since useless symbols do not occur in any terminal derivation.)

Let $G^c = (N, T, S, P^c)$ be a c.f. com. grammar, where $P^c = \{p_1, ..., p_s\}$. Further let $G = (N, T, S, \bar{P})$ be a c.f. grammar with $\bar{P} = \{\bar{p}_1, ..., \bar{p}_s\}$ such that G^c is the com. grammar induced by G. (G can be constructed in an obvious way.)

Let $\Pi = \{\pi_1, ..., \pi_s\}$ be the names of productions in P^c and $\bar{\Pi} = \{\bar{\pi}_1, ..., \bar{\pi}_s\}$ be the names of corresponding productions in $\bar{P}$. For any derivation D

$$D\colon \tilde{\alpha}_0 \underset{G^c}{\Longrightarrow} \tilde{\alpha}_1 \underset{G^c}{\Longrightarrow} \cdots \underset{G^c}{\Longrightarrow} \tilde{\alpha}_n$$

in G^c whose derivation word is π_D there is exactly one derivation

$$\bar{D}\colon \alpha_0 \underset{G}{\Longrightarrow} \alpha_1 \underset{G}{\Longrightarrow} \cdots \underset{G}{\Longrightarrow} \alpha_n$$

in G whose derivation word is $\pi_{\bar{D}}$ such that $\psi(\alpha_i) = \tilde{\alpha}_i$ for all $i = 1, ..., n$ and at each derivation step in D and $\bar{D}$ the corresponding productions are applied.

From this observation we make the following convention: The derivation tree of a derivation D in G^c means the derivation tree of the corresponding derivation $\bar{D}$ in G. Further, we identify π_i with $\bar{\pi}_i$ for all $i = 1,..., s$.

Now let us consider com. derivation words. Let $\psi: \Pi^* \to \Pi^{\oplus}$ be the Parikh mapping on Π^*. Then $\psi(D(G^c))$ denotes the com. image of $D(G^c)$ under ψ. For simplicity we shall use the following phrase: For a derivation tree Tr in G^c, "counting productions in Tr yields $\tilde{\pi} \in \Pi^{\oplus}$" means there is a derivation word π obtained from Tr such that $\psi(\pi) = \tilde{\pi}$.

Concerning the frontiers of derivation trees in G^c it is useful to define the following mapping χ. Let $[F(V)]$ denote the free abelian group generated by $V = N \cup T$. (A word in $[F(V)]$ has integer exponents in this case.) Define χ as

$$\chi: \Pi \to [F(V)]$$
$$\pi_i \mapsto A^{-1}\gamma,$$

where π_i is the name of the production $p_i = (A, \gamma)$, $i = 1,..., s$. We extend χ to a homomorphism from $\Pi^{\oplus}$ into $[F(V)]$ and denote it also by χ.

It is straightforward to see

FACT 2.1. *Let* Tr *be a derivation tree in G^c with root A and frontier $\alpha \in V^{\oplus}$. Also assume that counting productions in* Tr *yields $\tilde{\pi}$. Then it holds that*

$$\chi(\tilde{\pi}) = A^{-1} \cdot \alpha \in [F(V)].$$

In characterizing com. terminal derivation words we shall investigate terminal derivation trees in G^c. Such a tree may be arbitrarily large. Our observation is that if we work in G^c a terminal derivation tree may be regarded as the "sum" of some "minimal terminal derivation tree" and a finite number of "periodic derivation trees." We trees." We define these notions more precisely.

Let $G^c = (N, T, S, P^c)$ be as above. Let $P \subset P^c$ be a subset of P^c. P is said to be *connected* iff there is a terminal derivation D in G^c such that the set of productions applied in D is exactly P.

Consider a terminal derivation tree Tr in G^c. Let D_{Tr} be the leftmost derivation from Tr and P_{Tr} be the set of productions applied in D_{Tr}. Clearly, P_{Tr} is connected.

DEFINITION 2.2. Let Tr be a terminal derivation tree in G^c and $P \subset P^c$ be a subset of P^c. Tr is said to be *P-minimal* iff $P_{\mathrm{Tr}} = P$ and every $p \in P$ is applied at most $\#P+2$ times in any path of Tr without counting the production applied at the root.

(We label the edges of a derivation tree by production names in Π: the edges going out from a node are labeled by the name of the production applied at that node.)

It is straightforward to verify

FACT 2.3. *The depth of a P-minimal terminal derivation tree is bounded by*

$$\Theta(G^c) := (\#P^c + 1)^2.$$

DEFINITION 2.4. A derivation tree Tr in G^c is called *periodic* iff the following conditions hold:

(1) the label of the root of Tr is some nonterminal $A \in N$,

(2) exactly one leaf of Tr is labeled A, and

(3) all other leaves are labeled by terminals or ε.

We now give the characterization of com. terminal derivation words.

PROPOSITION 2.5. *For a com. word $\tilde{\pi} \in \Pi^{\oplus}$ let $P(\tilde{\pi})$ denote the set of productions whose names occur in $\tilde{\pi}$. Then $\tilde{\pi}$ is a com. terminal derivation word, i.e., $\tilde{\pi} \in \psi(D(G^c))$,* iff *the following conditions hold*:

(1) $\chi(\tilde{\pi}) = S^{-1}\tilde{w} \in [F(V)]$, *where* $\tilde{w} = t_1^{e_1} \cdots t_r^{e_r} \in T^{\oplus}$, $e_i \in \mathbb{N}_0$, $i = 1,..., r$.

(2) *There is a $P(\tilde{\pi})$-minimal terminal derivation tree* Tr_{π^*} *such that counting productions in* Tr *yields π^* and*[2] $\pi^* \leqslant \tilde{\pi}$.

In view of Proposition 2.5, let $\hat{\pi} \in \Pi^{\oplus}$ be a com. word such that $\pi^* + \hat{\pi} = \tilde{\pi}$. In proving this proposition we will see that from $\hat{\pi}$ periodic derivation trees can be constructed and inserted into Tr_{π^*} such that a terminal derivation tree for $\tilde{\pi}$ can be obtained. We need

LEMMA 2.6. *For every com. word $\delta \in \Pi^{\oplus}$ satisfying the condition*

$$\chi(\delta) = t_1^{e_1} \cdots t_r^{e_r} \in [F(V)], \qquad (\S)$$

[2] $\leqslant$ is defined on $\Pi^{\oplus}$ as follows: $x = \pi_1^{e_1} \cdots \pi_r^{e_r} \leqslant x' = \pi_1^{e'_1} \cdots \pi_r^{e'_r}$ iff $e_i \leqslant e'_i$ for all $i = 1,..., r$.

where $t_j \in T$, $e_j \in \mathbb{N}_0$, $j = 1, \ldots, r$, *there are finitely many periodic derivation trees and corresponding derivations in* G^c *of the form*

$$D_i \colon A_i \Rightarrow \tilde{u}_i A_i \tilde{v}_i,$$

where $A_i \in N$, $\tilde{u}_i, \tilde{v}_i \in T^{\oplus}$, $i = 1, \ldots, n$ *such that*

$$\sum_{i=1}^{n} \pi_{D_i} = \delta$$

(*and* $\sum_{i=1}^{n} \chi(\pi_{D_i}) = \sum_{i=1}^{n} (\tilde{u}_i + \tilde{v}_i) = t_1^{e_1} \cdots t_r^{e_r}$), *where* $\pi_{D_i} \in \Pi^{\oplus}$ *is the com. derivation word obtained from* D_i.

Proof of Lemma 2.6. We give a proof sketch and omit the formal details, which are left to the reader.

From δ we want to construct periodic derivation trees successively. The idea is as follows: Extract some production from δ, say $\bar{p}$. If $\bar{p}$ corresponds to a periodic derivation tree, then repeat the procedure with $\delta - \bar{p}$ instead of δ.

Otherwise, $\bar{p}$ is not a periodic derivation tree. We consider $\bar{p}$ and $\delta - \bar{p}$. We want to expand $\bar{p}$ by extracting productions from $\delta - \bar{p}$ until a periodic derivation tree is obtained. $\bar{p}$ is expanded as follows: Take a production from $\delta - \bar{p}$, say $\bar{p}'$, such that either

(1) the label of the root of $\bar{p}'$ is that of some leaf of $\bar{p}$, or

(2) some leaf of $\bar{p}'$ and the root of $\bar{p}$ have the same label.

Note that such a production ($\bar{p}'$) can always be obtained. Otherwise some nonterminal occurring in $\bar{p}$ would *not* appear in the remaining productions in $\delta - \bar{p}$, and therefore condition (§) was violated.

In case (1) expand $\bar{p}$ downward by replacing some leaf of $\bar{p}$ by $\bar{p}'$. In case (2) expand $\bar{p}$ upward by replacing some leaf of $\bar{p}'$ by $\bar{p}$. We now obtain a new tree formed by the expansion.

Repeat this expansion procedure with the new tree and $\delta - (\bar{p} + \bar{p})$ instead of $\bar{p}$ and $\delta - \bar{p}$ until a periodic derivation tree is obtained.

Thus by extracting productions from δ a periodic derivation tree can be constructed. If there are still productions which form the word $\delta' \in \Pi^{\oplus}$, then δ' also satisfies condition (§). Repeat the whole procedure with δ' instead of δ. After a finite number of times we obtain finitely many periodic derivation trees and corresponding derivations which satisfy the properties stated in the lemma. This completes the proof of Lemma 2.6. ∎

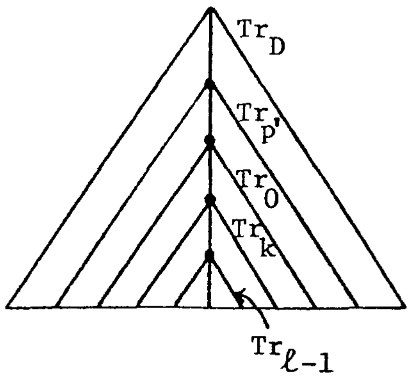
Proof of Proposition 2.5. *Only if* Let $\tilde{\pi} \in \psi(D(G^c))$. Then there is a terminal derivation D: $S \Rightarrow_{G^c}^{*} \tilde{w}$ for some $\tilde{w} \in L(G^c)$ such that $\psi(\pi_D) = \tilde{\pi}$, where π_D is the derivation word obtained from the derivation D. Further let

Tr_D be the corresponding derivation tree obtained from D. Condition 1 follows from Fact 2.1.

To show that condition 2 is also satisfied we look at the proof of Parikh's theorem (cf. Ginsburg, 1966, p. 146). Suppose that without counting the first derivation step no production of $P(\tilde{\pi})$ occurs more than $\#P(\tilde{\pi})+2$ times in any path of the tree Tr_D. Then Tr_D is a $P(\tilde{\pi})$-minimal terminal derivation tree with $\chi(\pi_D)=\tilde{\pi}$ and hence condition 2 is satisfied.

Now suppose that some production $p \in P(\tilde{\pi})$ with name π_λ is applied more than $\#P(\tilde{\pi})+2$ times in some path of the tree Tr_D (without counting the production at the root). Then for some production $p' \in P(\tilde{\pi})$ with name $\pi_{\lambda'}$ there must be a subtree $\mathrm{Te}_{p'}$ of Tr_D with the following property: On some path the subtree $\mathrm{Tr}_{p'}$ contains $l := \#P(\pi)+3$ edges $g_0,\ldots, g_{l-1}$ labeled by the same production name $\pi_{\lambda'}$, and no path of any subtree of $\mathrm{Tr}_{p'}$ contains more than $l-1$ edges labeled by the same production name.

Let $\mathrm{Tr}_0,\ldots, \mathrm{Tr}_{l-1}$ be the subtrees of $\mathrm{Tr}_{p'}$ such that one of the edges going out from the root of Tr_i, $i=0,\ldots, l-1$, is g_i. This situation is illustrated by



Let k be the least integer, $0 \leqslant k \leqslant l-2$, such that the productions occurring in Tr_k are exactly those occurring in Tr_{k+1}. Because the edges g_k and g_{k+1} are labeled by the same production name, the roots of the subtrees Tr_k and Tr_{k+1} are labeled by the same nonterminal. Therefore we can removed the subtree Tr_k from Tr_D and insert the subtree Tr_{k+1} at the node which was the root of Tr_k. The productions occurring in the resulting tree $\mathrm{Tr}_{D'}$ are $P(\tilde{\pi})$. Let $\pi_{D'}$ be some derivation word obtained from $\mathrm{Tr}_{D'}$ and $\tilde{\pi}_{D'} := \psi(\pi_{D'})$. Then we have $\tilde{\pi}_{D'} < \tilde{\pi}$. Now if $\mathrm{Tr}_{D'}$ is a $P(\tilde{\pi})$-minimal terminal derivation tree, we are done. Otherwise, we continue our procedure and ultimately obtain a $P(\tilde{\pi})$-minimal tree with the property stated in condition 2. This completes the proof of the only if part.

If Let $\tilde{\pi} \in \Pi^{\oplus}$. We show that if $\tilde{\pi}$ satisfies conditions 1 and 2, then $\tilde{\pi}$ is the com. image of some derivation word, i.e., $\tilde{\pi} \in \psi(D(G^c))$. Since condition 2 is satisfied, there is a $P(\tilde{\pi})$-minimal terminal derivation tree Tr_{π^*} such that counting the productions in Tr_{π^*} yields π^* and $\pi^* \leqslant \tilde{\pi}$. Let $\hat{\pi} := \tilde{\pi} - \pi^*$. It follows that

$$\chi(\hat{\pi}) = t_1^{e_1} \cdots t_r^{e_r} \in T^{\oplus}, \qquad e_i \in \mathbb{N}_0, \quad i = 1,..., r,$$

since $\pi^* \in \psi(D(G^c))$ and $\pi^* \leqslant \tilde{\pi}$.

Clearly, $\hat{\pi}$ satisfies condition (§) of Lemma 2.6. Therefore there are finitely many periodic derivation trees, denoted by $\mathrm{Tr}_1,..., \mathrm{Tr}_n$, and corresponding derivations of the form

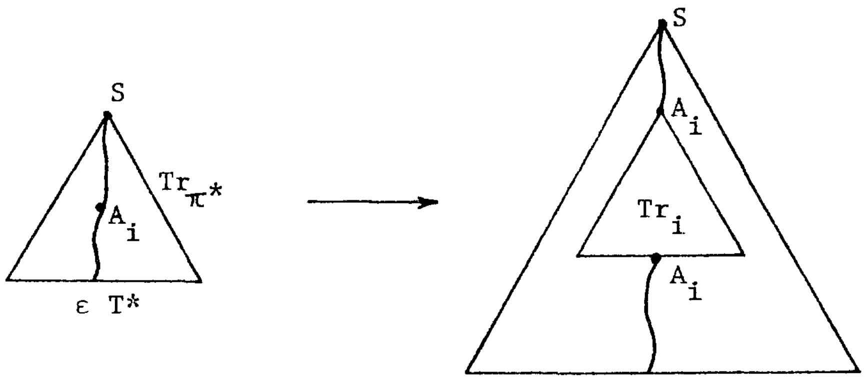
$$D_i\colon A_i \xrightarrow{*} \tilde{u}_i A_i \tilde{v}_i,$$

where $A_i \in N$, $\tilde{u}_i, \tilde{v}_i \in T^{\oplus}$, $i = 1,..., n$, such that

$$\sum_{i=1}^{n} \pi_{D_i} = \hat{\pi},$$

where π_{D_i} is the com. derivation word obtained from D_i.

The fact that Tr_{π^*} is $P(\tilde{\pi})$-minimal and each production applied in D_i is in $P(\tilde{\pi})$ implies the occurrence of the nonterminal A_i as label of some internal node of Tr_{π^*} for all $i = 1,..., n$. Therefore we can extend the tree Tr_{π^*} by inserting Tr_i at some internal node labeled by A_i in Tr_{π^*}



It is obvious that the set of productions occurring in the resulting derivation tree is $P(\tilde{\pi})$. Thus we can successively insert the derivation trees $\mathrm{Tr}_1,..., \mathrm{Tr}_n$ into Tr_{π^*} and obtain a terminal derivation tree $\mathrm{Tr}_{\tilde{\pi}}$ with the property that counting productions in $\mathrm{Tr}_{\tilde{\pi}}$ yields $\tilde{\pi}$. We conclude that $\tilde{\pi} \in \psi(D(G^c))$. This completes the proof of Proposition 2.5. ∎

2.2. *The Complexity of UWP-CFCG*

Having characterized com. terminal derivation words, we proceed to prove that UWP-CFCG is in *NP*. The following proposition is essential.

PROPOSITION 2.7. *Let $\tilde{w} \in T^{\oplus}$. Then $\tilde{w} \in L(G^c)$, iff there is a derivation word $\tilde{\pi} = \pi_1^{f_1} \cdots \pi_s^{f_s} \in \psi(D(G^c))$ such that*

(1) $\chi(\tilde{\pi}) = S^{-1}\tilde{w} \in [F(V)]$,

(2) $\sum_{i=1}^{S} f_i \leqslant e \cdot d^{\|c\|}$,

where $e = |\tilde{w}|$ and d is some fixed constant (independent of $\tilde{w}$ and G^c).

Proof. We first show Proposition 2.7 in a restricted form by assuming that G^c is ε-free and cycle-free (i.e., no production is of the form $A \to B$, where $A, B \in N$). Let G be the corresponding c.f. grammar. $\tilde{w} \in L(G^c)$ iff there is some terminal derivation tree Tr in G with frontier w such that $\psi(w) = \tilde{w}$. Tr contains no more than $2 \cdot e$ productions. Counting productions in Tr yields $\tilde{\pi} = \pi_1^{f_1} \cdots \pi_s^{f_s} \in \psi(D(G^c))$ and it holds that $\sum_{i=1}^{S} f_i \leqslant 2 \cdot e$. Hence Proposition 2.7 is proved in this case. (Note that the upper bound $2 \cdot e$ is achieved if G is in Chomsky normal form.)

We now consider the general case. We introduce some technical definitions. Concerning single productions in G we define the following notions.

Let $\mathrm{Tr} = \mathrm{Tr}(A, w)$ be a derivation tree with root $A \in N$ and frontier $w \in T^*$. Let $(u_1, u_2, ..., u_n)$ be a path in Tr, where u_1, u_n are internal nodes. $(u_1, ..., u_n)$ is called a *single path* iff for all $i = 1, ..., n-1$, (u_i, u_{i+1}) is labeled by names of single productions. A single path $(u_1, ..., u_n)$ is called a *cycle* iff u_1 and u_n are labeled by the same nonterminal. A derivation tree is called *cycle-free* iff it does not contain any cycle. Clearly, in a terminal derivation tree cycles may be removed without affecting the frontier. Therefore, in order to obtain a terminal derivation tree for $w \in L(G)$ it suffices to consider cycle-free terminal derivation trees.

Concerning ε-productions in P we consider the set

$$N_\varepsilon := \{A \in N : A \overset{*}{\Longrightarrow} \varepsilon\} \subset N.$$

A derivation tree $\mathrm{Tr}(A, \varepsilon)$ with root A and frontier ε is called an ε-tree. An ε-*subtree* of a terminal derivation tree is a subtree which is an ε-tree. An ε-subtree Tr′ of some terminal derivation tree Tr is said to be *maximal* if there is no ε-subtree of Tr whose root is a proper ancester of the root of Tr′.

The proof idea is as follows. In order to obtain a terminal derivation tree for $w \in L(G)$ it suffices to search for a cycle-free terminal derivation tree whose maximal ε-subtrees are small.

Let $w \in L(G)$ and Tr be a terminal derivation tree with frontier w such that $\psi(w) = \tilde{w}$. Consider a maximal ε-subtree $\mathrm{Tr}(A, \varepsilon)$ of Tr such that counting productions in $\mathrm{Tr}(A, \varepsilon)$ yields $\tilde{\delta}$. In view of Proposition 2.5 we want to replace $\mathrm{Tr}(A, \varepsilon)$ in Tr by a "small" ε-tree with root A. Indeed, we may replace $\mathrm{Tr}(A, \varepsilon)$ by any ε-tree with root A which is chosen as small as possible. The resulting tree still has frontier w. Thus we may choose an ε-tree such that no nonterminal appears more than once in any path of that tree. Call such an ε-tree a *minimal* ε-tree.

Let Tr_+ be the tree obtained in the following way: Delete first all maximal ε-subtrees in Tr including the roots of these subtrees and then all cycles. The productions in Tr_+ are exactly productions in G', the ε-free c.f. grammar equivalent to G, which is constructed by the well-known method (cf. Ginsburg, 1966, p. 38).

We now estimate the size of Tr_+, which is "small." By "inserting" appropriate minimal ε-trees into Tr_+ we then obtain a "small" terminal derivation tree with frontier w.

First note that the number of single productions in G' (the ε-free c.f. grammar equivalent to G) is $\leqslant (\#N)^2$. As observed at the beginning of the proof, we have:

(1) The number of non-single productions in Tr_+ is $\leqslant 2e$. Concerning the number of single paths in Tr_+, we have:

(2) The number of single paths in Tr_+ is $\leqslant 2e$. (Since there are $2e$ nodes in a binary tree with e leaves.)

A single path in Tr_+ contains no cycles. Therefore, each single path in Tr_+ contains at most $(\#N - 2)$ single productions. Thus there are at most $2e(\#N - 2)$ single productions in Tr_+. (Note that these productions are either productions in G or new productions obtained from the construction of G'.)

Hence the number of productions in Tr_+ is bounded by

$$2e + 2e(\#N - 2) = 2e(\#N - 1).$$

It is not hard to see that there is to this tree Tr_+ a tree $\overline{\mathrm{Tr}}$ in G which satisfies (3) and (4):

(3) There is a 1–1 correspondence between productions in Tr_+ and $\overline{\mathrm{Tr}}$,

(4) The frontier of $\overline{\mathrm{Tr}}$ is a word in $(T \cup B_\varepsilon)^*$ such that deleting nonterminals yields w.

Let l denote the maximal length of the right-hand sides of productions in P. Then the number of nonterminal leaves of $\overline{\mathrm{Tr}}$ is bounded by $(l - 1) \cdot$

$2e(\#N-1)$. For each nonterminal leaf we insert an appropriate minimal ε-tree and obtain a terminal derivation tree with frontier w. Thus the resulting tree has at most

$$2e(\#N-1)+h\cdot(l-1)\cdot 2e(\#N-1)=2e(\#N-1)[h(l-1)+1]$$

productions, where h is the largest number of productions in a minimal ε-tree.

Recall that in a minimal ε-tree no nonterminal appears more than once in any path. Therefore

$$h \leqslant l^{\#N}.$$

A simple computation shows the upper bound in (2). This completes the proof of this proposition. ▮

We are now able to prove that UWP-CFCG is in *NP*.

Proposition 2.8. UWP-CFCG *is in NP.*

Proof. To show that UWP-CFCG is in *NP* we apply Proposition 2.7. According to this proposition there is to $\tilde{w} \in L(G^c)$ a com. derivation word $\tilde{\pi} \in \psi(D(G^c))$ which satisfies conditions (1) and (2). Condition (2) means that the size of $\tilde{\pi}$ written as a com. word is polynomially bounded.

Verify that $\tilde{w} \in L(G^c)$ as follows. Nondeterministically guess in polynomial-time a com. word $\tilde{\pi} \in \Pi^{\oplus}$ of polynomial-bounded size. Then check the following properties of $\tilde{\pi}$:

(1) $\chi(\tilde{\pi}) \overset{?}{=} S^{-1}\tilde{w}$. This can be done deterministically in polynomial-time, since it is exactly the problem of determining solvability of linear diophantine equation systems which is known to be in nondeterministic polynomial time.

(2) $\tilde{\pi} \overset{?}{\in} \psi(D(G^c))$. By Proposition 2.5 we have only to check the second condition of this proposition: nondeterministically guess a $P(\tilde{\pi})$-minimal terminal derivation tree Tr_{π^*} in polynomial time such that counting productions in Tr_{π^*} yields π^* and $\pi^* \leqslant \tilde{\pi}$. Actually we do not produce an explicit representation of Tr_{π^*} which may be exponentially large. We gues Tr_{π^*} top–down and level by level. At each level we store the com. sentential form obtained so far whose size is polynomially bounded. There are at most $\#P(\tilde{\pi})$ productions which are applicable at a level, and the multiplicity of the application of a production is guessed nondeterministically (since there may be distinct productions with the same nonterminal on the left-hand side). Note that we also recors the total number of productions guessed. When the bottom level is reached, we obtain π^*. Thus checking wether $\tilde{\pi} \in \overset{?}{\in} \psi(D(G^c))$ can be done nondeterministically in polynomial time.

There $\tilde{w} \overset{?}{\in} L(G^c)$ can be tested nondeterministically in polynomial time. This completes the proof of Proposition 2.8. ∎

PROPOSITION 2.9. UWP-CFCG *is log-hard for NP.*

Proof. This fact follows from the *NP*-hardness of the UWP-RCG, which will be shown in the next section. ∎

From Propositions 2.8 and 2.9 we obtain

THEOREM 2.10. UWP-CFCG *is log-complete for NP.*

Remark 2.11. In van Leeuwen (1974) it has been shown that the arcone reachability problem for a subclass of vector addition systems is decidable (cf. van Leeuwen, 1974, Theorem 3.2). In our notations this subclass is the class of c.f. com. grammars.

3. THE COMPLEXITY OF THE UNIFORM WORD PROBLEM FOR REG. COM. GRAMMARS AND FOR RATIONAL EXPRESSIONS IN FREE COM. MONOIDS

Consider first rational expressions in free com. monoids with bounded star height. The height of a rational expression is defined recursively as follows. Let V be a finite alphabet. M denotes the com. monoid $V^{\oplus}$. Define the star height of rational expressions as a mapping h from the set of rational expressions into $\mathbb{N}_0$:

$$h(\varnothing) := 0,\ h(m) := 0 \text{ for all } m \in M,$$
$$h(\alpha_1 \circ \alpha_2) := \max\{h(\alpha_1), h(\alpha_2)\},\ \circ \in \{\cdot, \cup\},$$
$$h(\alpha^*) := h(\alpha) + 1,$$

where α, α_1, α_2 are rational expressions.

The uniform word problem for rational expressions in M with star height $\leqslant k$, $k \in \mathbb{N}_0$, is defined as follows: Given a com. word $w \in M$ and a rational expression α in M with $h(\alpha) \leqslant k$ it is to decide whether $w \in L(\alpha)$. This problem is denoted by UWP-RE(V, k).

PROPOSITION 3.1. UWP-RE($\{a\}, 0$) *is log-complete for NP.*

Proof. Consider the membership problem for integer expressions, denoted by $\mathbb{N}_0$-MEMBER. (Integer expressions are expressions involving non-negative integers written in binary with two binary operations: union and addition. Integer expressions denote subsets of $\mathbb{N}_0$. (The interested reader is referred to Stockmeyer and Meyer, 1973, for details.))

It can be easily seen that $\mathbb{N}_0$-MEMBER and UWP-RE($\{a\}, 0$) are polynomially equivalent: com. words in $\{a\}^{\oplus}$ represent nonnegative integers and vice versa; the operations $\cup, \cdot$ in rational expressions in $\{a\}^{\oplus}$ are exactly the operations, $\cup, +$ in tnteger expressions.

Since $\mathbb{N}_0$-MEMBER is log-complete for *NP* (cf. Stockmeyer and Meyer, 1973), Proposition 3.1 follows. ∎

COROLLARY 3.2. *For any* $k \in \mathbb{N}_0$: UWP-RE(V, k) *is log-complete for NP.*

Proof. Rational expressions in free com. monoids can be simulated by c.f. com. grammars. This proves the upper bound. *NP*-hardness follows from Proposition 3.1. ∎

We have the following corollaries.

COROLLARY 3.3. *The uniform word problem for rational expressions in free com. monoids is log-complete for NP.*

COROLLARY 3.4. UWP-RCG *is log-complete for NP.*

4. The Complexity of the Uniform Word Problem for Context-Sensitive Commutative Grammars

We show that UWP-CSCG is log-complete for PSPACE.

PROPOSITION 4.1. *The uniform word problem for context-sensitive com. grammars is log-complete for* PSPACE.

Proof. We first show that UWP-CSCG is in PSPACE. Let $G^c = (N, T, S, P^c)$ be a c.s. com. grammar and $w \in T^{\oplus}$, $w = t_1^{e_1} \cdots t_r^{e_r}$, $e_i \in \mathbb{N}_0$, $i = 1,..., r$. From Definition 1.2 we have

$$\forall p = (\alpha, \beta) \in P^c : |\alpha| \leqslant |\beta|.$$

Consider the known construction of an equivalent nondeterministic linear bounded automaton for a c.s. grammar. In a similar way we can construct a nondeterministic linear bounded automaton accepting $L(G^c)$. We leave the construction to the reader.

To show that UWP-CSCG is PSPACE-hard we reduce the word problem for nondeterministic linear bounded automata to UWP-CSCG. The construction is similar to that in Jones *et al.* (1977) and is omitted.

Thus UWP-CSCG is log-complete for PSPACE. ∎

TABLE 1

The Uniform Word Problem for	Upper Bound	Lower Bound
Rational expressions with bounded star height	*NP*	Log-complete
Rational	*NP*	Log-complete
Regular commutative grammars	*NP*	Log-complete
Context-free commutative grammars	*NP*	Log-complete
Context-sensitive commutative grammars	PSPACE	Log-complete
Type 0 commutative grammars	Decidable	EXSPACE

5. Concluding Remarks

In this paper we have introduced commutative grammars and classified the complexity of the uniform word problem for c.s., c.f., reg. com. grammars and for rational expressions in free com. monoids. Concerning type 0 com. grammars we do not have an exact classification yet. The decidability of the uniform word problem for type 0 com. grammars has been announced in Mayr (1981). On the other hand, and EXSPACE lower bound can be obtained from a result by Cardoza, Lipton, and Meyer. They showed that the uniform word problem for commutative Thue systems is EXSPACE-complete. (The interested reader is referred to Mayr and Meyer (1982) for a more detailed presentation of this result.)

We summarize the results in Table 1.

Acknowledgments

The author wishes to thank Professor G. Hotz for many suggestions, B. Becker and K. Estenfeld for reading the manuscript, and an anonymous referee for helpful advice in clarifying the presentation of this paper.

Received: June 25, 1982; revised: August 6, 1983

References

Crespi-Reghizzi, S. and Mandrioli, D. (1976), Commutative grammars, *Estratto da Calcolo* **XIII**, 173–179.

Eilenberg, S. and Schützenberger, M. P. (1969), Rational sets in commutative monoids, *J. Algebra* **13**, 173–191.

Ginsburg, S. (1966), "The Mathematical Theory of Context-Free Languages," McGraw–Hill, New York.

Hotz, G. (1980), Eine Neue Invariante für Kontext-Freie Sprachen, *Theoret. Comput. Sci.* **11**, 107–116.

HUNT, H., ROSENKRANTZ, D., AND SZYMANSKI, T. (1976), On the equivalence, containment and covering problems for the regular and context-free languegs, *J. Comput. System Sci.* **12**, 222–268.

JONES, N., LANDWEBER, L., AND LIEN, Y. 1977), Complexity of some problems in Petri nets, *Theoret. Comput. Sci.* **4**, 227–299.

VAN LEEUWEN, J. (1974), A partial solution to the reachability problem for vector addition systems, *in* "Proceeding of the 6th Annual STOC," pp. 303–307.

MAYR, E. AND MEYER, A. (1982), The complexity of the word problems for commutative semigroups and polynomial ideals, *Advan. in Math.* **46**, 305–329.

MAYR, E. (1981), An algorithm for the general Petri net reachability problem, *in* "Proceeding of the 13th Annual STOC," pp. 238–246.

STOCKMEYER, L. AND MEYER, Z. (1973), Word problems requiring exponential time, *in* "Proceeding, of the 5th Annual STOC," pp. 1–9.