Theoretical Computer Science 33 (1984) 305-326
North-Holland

DECIDING THE INEQUIVALENCE OF CONTEXT-FREE GRAMMARS WITH 1-LETTER TERMINAL ALPHABET IS Σ_2^p-COMPLETE

Dung T. HUYNH*
Fachbereich Informatik, Universität des Saarlandes, D-6600 Saarbrücken, Fed. Rep. Germany

Communicated by R. Book
Received May 1982
Revised May 1983

Abstract. This paper deals with the complexity of context-free grammars with 1-letter terminal alphabet. We study the complexity of the membership problem and the inequivalence problem. We show that the first problem is NP-complete and the second one is Σ_2^p-complete with respect to log-space reduction. The second result also implies that the inequivalence problem is in PSPACE, solving an open problem stated by Hunt, Rosenkrantz and Szymanski (1976).

Introduction

One of the important research topics in "Complexity Theory" is the investigation of the computational complexity of decision problems in various areas of mathematics and computer science. Decision problems in automata theory and formal language theory are of special interest, and have been extensively studied by many authors (cf. [3, 8, 9, ...]).

The two basic decision problems in formal language theory, namely the membership problem and the equivalence problem, concerning regular expressions, extended regular expressions, various classes of grammars have been studied first by Meyer, Stockmeyer, Hunt, Rosenkrantz, Szymanski, ...

In this paper we are also concerned with these decision problems in connection with context-free grammars generating 1-letter alphabet languages. In this case there is no difference between the equivalence problem and the commutative equivalence problem, which has been studied by the present author [5, 6]. (The reason for our study on the complexity of commutative equivalence problems is that commutativity provides us necessary criteria for testing inequivalence of grammars. The complexity of these criteria is in general much lower.)

We shall show that for this class of grammars the membership problem is log-complete for NP and the inequivalence problem is log-complete for Σ_2^p, the

* Present address: Computer Science Department, Iowa State University, Ames, IA 50011, U.S.A.

second level of the polynomial-time hierarchy introduced by Meyer and Stockmeyer [9].

Since $\Sigma_2^P \subset$ PSPACE, we have the fact that this equivalence problem is in PSPACE, solving an open problem stated by Hunt, Rosenkrantz and Szymanski [3]. This is also the main result of this paper (cf. the discussion at the end of Section 1)

This paper is organized as follows. In Section 1 we review well-known definitions used in our research. Section 2 deals with the classification of the complexity of the membership problem. In Section 3 we derive some properties concerning the commutative images of context-free languages, which are necessary in order to show that context-free 1-letter alphabet languages can be expressed as ultimately periodic sets by 'small' representations (cf. Section 4). The latter fact will be proved in Section 4. In Section 5 we classify the complexity of the inequivalence problem. We conclude this paper by some remarks in Section 6.

1. Preliminaries and results

In this section we review commonly known definitions and give notations which will be used later.

For a finite alphabet Σ, let Σ^* denote the free monoid generated by Σ, and Σ^+ denote the free semigroup generated by Σ. $\varepsilon \in \Sigma^* \backslash \Sigma^+$ denotes the empty word. A subset $L \subset \Sigma^*$ is called a language. A language $L \subset \Sigma^*$ with $\text{card}(\Sigma) = 1$ is called a *1-letter alphabet language* (or 1LA language for short), where $\text{card}(S)$ denotes the cardinality of the set S.

In the whole paper, $G = (N, T, S, P)$ denotes a context-free grammar (c.f. grammar for short), where N is the set of nonterminals, T the set of terminals, $S \in N$ is the axiom and $P \subset N \times (N \cup T)^*$ is the finite set of productions. The language generated by G is denoted by $L(G)$. The relations $\Rightarrow_G, \Rightarrow_G^*, \Rightarrow_G^+$ are defined as usual. We often write $\Rightarrow, \Rightarrow^*, \Rightarrow^+$ if G is understood.

In this paper we are concerned with 1LA languages. A c.f. grammar generating an 1LA language is called a *c.f. 1-letter terminal alphabet grammar* (or c.f. 1LTA grammar for short).

The main decision problem studied in this work is the inequivalence problem for c.f. 1LTA grammars. In this case it holds that two c.f. 1LTA grammars G_1, G_2 are equivalent iff they are commutative equivalent, i.e., iff they have the same commutative image.

For the proof of the upper bound of the complexity of this decision problem we shall investigate some properties of the commutative images of c.f. languages. For this purpose we introduce the notions 'semilinear sets' and 'representations' of similinear sets for the sake of completeness.

In the following, $\mathbb{N}_0$ denotes the set of nonnegative integers, $\mathbb{N}$ denotes the set of positive integers and $\mathbb{Z}$ the set of integers.

Definition 1.1. Let C and Π be two finite subsets of $\mathbb{N}_0^k$ and let $C \neq \emptyset$. Then $L(C;\Pi)$ denotes the following set:

$$L(C;\Pi) := \left\{ c + \sum_{\pi \in \Pi} \lambda_\pi \pi \mid c \in C \text{ and } \lambda_\pi \in \mathbb{N}_0 \right\}.$$

If $C = \{c_1, \ldots, c_r\}$ and $\Pi = \{\pi_1, \ldots, \pi_s\}$, we also write

$$L(c_1, \ldots, c_r; \pi_1, \ldots, \pi_s).$$

Further, if $C = \emptyset$, then $L(C;\Pi) := \emptyset$.

A subset $L \subset \mathbb{N}_0^k$ is called a *linear set* iff $L = L(C;\Pi)$ for some finite subsets C and Π with $C = \{c\}$. c is called the *constant* of L, Π the *period system* of L. An element $\pi \in \Pi$ is called a *period* of L.

A subset $SL \subset \mathbb{N}_0^k$ is called a *semilinear set* (s.l. set for short), iff SL is a finite union of linear sets.

If $L = L(c;\Pi)$ is a linear set, we call $(c;\Pi)$ a *representation* of L. Obviously, a linear set can have different representations. The constants of such representations must be the same. But the period systems may be different.

If $SL = L(c_1;\Pi_1) \cup \cdots \cup L(c_m;\Pi_m)$ is a s.l. set, then $(c_1;\Pi_1), \ldots, (c_m;\Pi_m)$ is called a *representation* of SL. Two representations $(c_1;\Pi_1), \ldots, (c_m;\Pi_m)$ and $(\bar{c}_1;\bar{\Pi}_1), \ldots, (\bar{c}_n;\bar{\Pi}_n)$ are said to be *equivalent* if they define the same s.l. set.

For a finite alphabet $U = \{a_1, \ldots, a_k\}$ define the following mapping:

$$\psi_U : U^* \to \mathbb{N}_0^k$$

$$w \mapsto \psi_U(w) := (|w|_{a_1}, |w|_{a_2}, \ldots, |w|_{a_k}),$$

where $|w|_{a_i}$ denotes the number of the occurrences of a_i in the word w, $i = 1, \ldots, k$. ψ_U is called the *Parikh-mapping*. We often write ψ instead of ψ_U if the alphabet U is understood.

For a language $L \subset U^*$, $\psi(L)$ is called the *commutative image* of L.

We shall characterize the complexity of problems in terms of known complexity classes. We assume that the reader is familiar with the basic notions from complexity theory, for instances P, NP, 'log-complete', 'log-hard', 'log-space computable', ... Further, the polynomial-time hierarchy introduced by Meyer and Stockmeyer (cf. [9]) is denoted by

$$\Sigma_0^P \subset \Sigma_1^P \subset \Sigma_2^P \subset \cdots \subset \Sigma_k^P \subset \cdots, \qquad \Pi_0^P \subset \Pi_1^P \subset \Pi_2^P \subset \cdots \subset \Pi_k^P \subset \cdots,$$

where $\Sigma_0^P = \Pi_0^P = P$ and $\Sigma_1^P = NP$, $\Pi_1^P = \text{co-NP}$. For an exact definition, the reader is referred to [9].

We now give the definitions of the decision problems studied in this paper.

Definition 1.2. *The membership problem for c.f. 1LTA grammars, denoted by*

MEMBER: Given a c.f. 1LTA grammar $G = (N, \{0\}, S, P)$ and a non-negative integer $n \in \mathbb{N}_0$, determine whether $0^n \in L(G)$.

The inequivalence problem for c.f. 1LTA grammars, denoted by

INEQ: Given two c.f. 1LTA grammars G_1, G_2 with the same terminal alphabet, determine whether $L(G_1) \neq L(G_2)$.

It is straightforward to formulate MEMBER and INEQ as languages. The reader shoud note that, as regards MEMBER, the integer n is encoded in binary notation.

Main Results. (1) MEMBER *is log-complete for* NP.
(2) INEQ *is log-complete for* Σ_2^p.

Remark 1.3. The first combinatorial problem shown to be complete for Σ_2^p is the inequivalence problem for integer expressions. This is proved by Meyer and Stockmeyer [9]. Recently, the present author has found some new decision problems complete for this class of the polynomial-time hierarchy, e.g., the inequivalence problem for semilinear sets. The proof of this result can be found in [4].

In proving the above results the main difficulty is to obtain the upper bound for INEQ. It is well known that c.f. 1LTA languages are regular. A representation of a c.f. 1LTA language as an ultimately periodic set can be constructed by the procedure implied by the proof of this fact (cf. [2, p. 86]). But this approach gives us an exponential upper bound.

One may use another method implied by the proof of Parikh's theorem. Consider the commutative images of c.f. 1LTA languages and apply the result in [4] for the inequivalence of s.l. sets. This approach also provides us an exponential upper bound.

The idea of our proof is as follows. We also consider representations of c.f. 1LTA languages as ultimately periodic sets. To obtain 'small' representations we shall refine the proof of Parikh's theorem so that a property between the constants and periods of the semilinear representation of the commutative image of a c.f. language can be derived. With this property we will be able to prove that a small representation of a c.f. 1LTA language as an ultimately periodic set exists. This provides us the desired upper bound for INEQ.

2. The complexity of the membership problem

In this section we classify the complexity of MEMBER.

Proposition 2.1. MEMBER *is in* NP.

Proof. In [5] it has been shown that the uniform word problem for c.f. commutative grammars is NP-complete. To show that MEMBER is in NP we reduce it to this problem.

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

(1) N, T are finite sets, $N \cap T = \emptyset$,

(2) $S \in N$,

(3) $P^c \subset N \times (N \cup Y)^{\oplus}$ is a finite set, where $M^{\oplus}$ denotes the free commutative monoid generated by M.

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

Note that in this case we work on free commutative monoids rather than on free monoids. Sentential forms are elements of $(N \cup T)^{\oplus}$, the free commutative monoid generated by $N \cup T$. (For more details, the interested reader is referred to [5].)

The uniform word problem for c.f. commutative grammars, denoted by UWP-CFCG as in [5], is defined as follows: Given a word $w \in T^{\oplus}$ and a c.f. commutative grammar $G^c = (N, T, S, P^c)$ determine whether w is in the language generated by G^c. (A language is a subset of a free commutative monoid in this case.)

Note that a word $u \in \{v_1, \ldots, v_r\}^{\oplus}$ is written as $u = v_1^{e_1} \ldots v_r^{e_r}$, $e_i \in \mathbb{N}_0$, $i = 1, \ldots, r$, where e_i is the number of the occurrences of v_i in u. Thus, if written on a Turing machine tape, u is encoded by the binary representation of e_i following the code of v_i, $i = 1, \ldots, r$.

It has been shown in [5] that UWP-CFCG is NP-complete. If the input grammar of an instance of MEMBER is written as a commutative grammar, MEMBER is a special case of UWP-CFCG in the sense that the grammar has a one-letter terminal alphabet. Therefore, we conclude that MEMBER is in NP. □

Proposition 2.2. MEMBER *is log-hard for* NP.

Proof. We construct a log-space reduction from the knapsack problem to MEMBER.

The knapsack problem is defined as follows:

$$\text{KNAPSACK} := \{(\text{bin}(a_1), \ldots, \text{bin}(a_n); \text{bin}(b)) \mid a_1, \ldots, a_n, b \in \mathbb{N}_0 \text{ and there is a subset } I \subset \{1, \ldots, n\} \text{ such that } \textstyle\sum_{i \in I} a_i = b\},$$

where $\text{bin}(a)$, $a \in \mathbb{N}_0$, denotes the binary representation of a without leading zeros.

We construct a reduced c.f. grammar $G = (N, T, S, P)$ such that

$$(\text{bin}(a_1), \ldots, \text{bin}(a_n); \text{bin}(b)) \in \text{KNAPSACK} \Leftrightarrow (G, b) \in \text{MEMBER}, \qquad (\bigstar)$$

where G depends on $a_1, \ldots, a_n$.

The only difficulty is to describe integers in binary representation by not too many c.f. productions.

Let m be the maximal length of $\text{bin}(a_1), \ldots, \text{bin}(a_n)$. Consider the productions

$$(!) \qquad B_m \to B_{m-1} B_{m-1}, \ldots, B_2 \to B_1 B_1, B_1 \to 0.$$

Clearly, $B_m \Rightarrow^* 0^{2^m}$.

Now let $A_1, \ldots, A_n$ be new nonterminals. For each i, $i = 1, \ldots, n$, let $i_1, \ldots, i_k$, $1 \leq i_1 < \cdots < i_k \leq m$, denote the positions of the digit 1 in $\text{bin}(a_i)$. Define

(2) $\quad A_i \to B_{i_1} \ldots B_{i_k}, \quad i = 1, \ldots, n \ (A_i \to \varepsilon, \text{ if } a_i = 0).$

Then P consists of the productions from (1), (2) and the following ones

$$S \to X_1 \ldots X_n, \quad X_i \to A_i \mid \varepsilon, \ i = 1, \ldots, n.$$

Obviously, (★) holds.

On the other hand, the reduction constructed is easily seen to be computable in log-space. Thus Proposition 2.2 is proved. □

From Proposition 2.2 we obtain the following.

Theorem 2.3. MEMBER *is log-complete for* NP.

Corollary 2.4. *The membership problem for c.f. IITA grammars generating finite languages is log-complete for* NP.

3. Some observations on the commutative images of c.f. languages

In this section we shall make some observations on the commutative images of c.f. languages. It is well known that the commutative images of c.f. languages are s.l. sets, as stated in Parikh's theorem (cf. [2, p. 146]). The proof of this theorem also provides us an effective procedure for computing a representation of the commutative image of the language generated by a given c.f. grammar.

In the following, let $G = (N, T, S, P)$ denote a c.f. grammar. W.l.o.g. we assume that G is reduced.

Let $L_1 \cup \cdots \cup L_m$, $L_i = L(c_i; \Pi_i)$, $i = 1, \ldots, m$ such that $(c_1; \Pi_1), \ldots, (c_m; \Pi_m)$ is the representation of $\psi(L(G))$ computed by the procedure implied by the proof of Parikh's theorem (cf. [2, p. 146]). Our aim is to obtain a property between the constants c_i's and period systems Π_i's. More precisely, we shall prove that for a certain constant c_i there is some j and $\pi_1, \ldots, \pi_r \in \Pi_j$ such that $c_i + \pi_1 + \cdots + \pi_r$ belongs to $L(c_j; \Pi_j)$.

A precise statement of this property is contained in Lemma 3.11, which will be applied in Section 4 to obtain a 'small' representation of $\psi(L(G))$ as an ultimately periodic set, provided T is a single-letter alphabet. Lemma 3.11 will be proved by a detailed analysis of the proof of Parikh's theorem.

For this purpose we first introduce the notion 'reachability graph' of a c.f. grammar and derive some properties induced by such graphs. We then give a refined characterization of $\psi(L(G))$ and prove the desired property between the constants c_i's and period systems Π_i's.

3.1. The reachability graph of a c.f. grammar

Let G be as above. Let $V := N \cup T$, $V_\varepsilon := V \cup \{\varepsilon\}$.

Definition 3.1. The *reachability graph* of G, denoted by $\Gamma(G)$, is the bipartite digraph $\Gamma(G) = (W, F)$ satisfying the following conditions:

(i) $W := V_\varepsilon \cup P$ is the set of vertices, where V_ε and P are the two disjoint subsets of W.

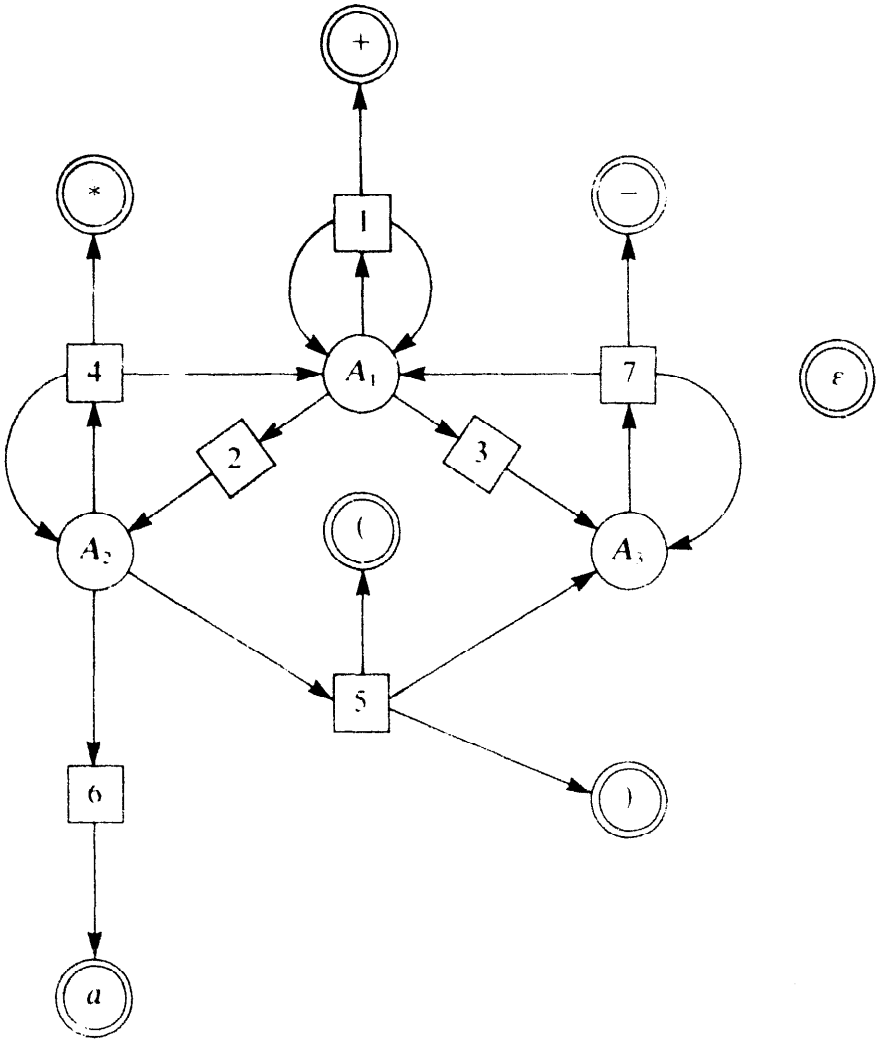
(ii) The set F of edges of $\Gamma(G)$ is defined as follows. Let $p \in P$ be a production. We have two cases:

- $p = (A \to \varepsilon)$ is an ε-production: $(A, p), (p, \varepsilon)$ are edges in F.
- $p = (A \to X_1 \dots X_n)$, $n \geq 1$: the edges $(A, p), (p, X_1), \dots, (p, X_n)$ are edges in F.

Example. The reachability graph $\Gamma(G_1)$ of the c.f. grammar G_1 with $N = \{A_1, A_2, A_3\}$, $T = \{+, -, *, a, (,)\}$, $S = A_1$ and productions

$$A_1 \to A_1 + A_1 \mid A_2 \mid A_3, \qquad A_2 \to A_2 * A_1 \mid (A_3) \mid a, \qquad A_3 \to A_1 - A_3,$$

which are denoted by $p_1, p_2, \dots, p_7$ respectively, is the following bipartite digraph:



where a vertex p_i from P is denoted by $\boxed{i}$, a vertex from N by $\textcircled{A_i}$ and a vertex from $T \cup \{\varepsilon\}$ by $\textcircled{@}$.

Remark 3.2. Note that we allow $\Gamma(G)$ to have parallel edges. From the definition of $\Gamma(G)$ it is easy to see that all the nonterminals of the above grammar G_1 are reachable, since it holds that

$$\exists \alpha, \beta \in V^*: A \Rightarrow^+ \alpha B \beta$$

iff there is a ditected path in $\Gamma(G)$ which leads from A to B.

The following fact holds trivially.

Fact 3.3. *If $L(G)$ is infinite, then $\Gamma(G)$ contains circuits.*

For our purpose it is important to introduce the following relation on W ($\Gamma(G) = (W, F)$). Define for all $v, v' \in W$: $v \sim v'$, iff

- either $v = v'$,
- or there is a directed path from v to v' and there is also a directed path from v' to v.

Clearly, $\sim$ is an equivalence relation and it partitions W into equivalence classes. Consider these equivalence classes. It is obvious that $[t]$ contains exactly one vertex for all $t \in T \cup \{\varepsilon\}$. For our purpose, such equivalence classes are not interesting. We only consider equivalence classes which contain at least two vertices from $N \cup P$. It is straightforward that these equivalence classes have at least one vertex from N and one vertex from P. We call such an equivalence class a *strongly-connected component* of $\Gamma(G)$ or simply a *component*.

Let $[A]$ be a component of $\Gamma(G)$. Then there is at least one circuit of length ≥ 2 through A and hence $A \Rightarrow^+ \alpha A \beta$ holds for some $\alpha, \beta \in V^*$. A nonterminal $A \in N$ wiith $A \Rightarrow^+ \alpha A \beta$ for some $\alpha, \beta \in V^*$ is called *recursive*. Otherwise, it is called nonrecursive.

Fact 3.4. *Let* $\mathrm{Rec}(G)$ *denote the set of recursive nonterminals of* G. *Further let* $\mathrm{Non}([A])$ *denote the vertices of the component* $[A]$ *which are in* V. *Then it holds that*

$$\mathrm{Rec}(G) = \bigcup_{\substack{[A] \text{ is a component} \\ \text{of } \Gamma(G)}} \mathrm{Non}([A]).$$

Definition 3.5. Let $\emptyset \neq N' \subseteq \mathrm{Rec}(G)$. N' is called *p-consistent*, if the following conditions hold:

(i) There is a component $[A]$ of $\Gamma(G)$ with $N' \subseteq [A]$.

(ii) For every vertex $B \in N' \subseteq [A]$ there is a directed circuit $(B, p_{i_1}, B_{i_1}, \ldots, p_{i_r}, B_{i_r}, p_{i_{r+1}}, B)$ with $B_{i_1}, \ldots, B_{i_r} \in N'$

Remark 3.6. From Definition 3.5 it follows that for each component $[A]$ of $\Gamma(G)$ the set $\mathrm{Non}([A])$ is p-consistent. In the next subsection we will see that p-consistent sets are exactly those sets from which the period systems of $\psi(L(G))$ can be constructed. (Hence we call them p- (period-)consistent.) Clearly, not all subsets of $\mathrm{Rec}(G)$ which are contained in some component $[A]$ of $\Gamma(G)$ are p-consistent.

3.2. A refined characterization of $\psi(L(G))$

In this subsection we shall investigate the structure of terminal derivation trees of G. This investigation will provide us a refined characterization of the commutative images of c.f. languages. We need some definitions. In the following we identify a component $[A]$ with the set of vertices labeled by nonterminals.

Definition 3.7. Let $N' \subset \mathrm{Rec}(G)$ be a p-consistent set and Tr be a terminal derivation tree. We say that N' *occurs* in Tr or Tr *contains* N', if every element of N' occurs as a node label in Tr.

Let $\boldsymbol{C}(G)$ denote the set of components of $\Gamma(G)$ and $\mathscr{P}(\boldsymbol{C}(G))$ denote the power set of $\boldsymbol{C}(G)$. An element $\theta \in \mathscr{P}(\boldsymbol{C}(G))$ is called *p-admissible*, if there is a terminal derivation tree Tr such that each element in θ occurs in Tr.

We denote the set of p-admissible elements from $\mathscr{P}(\boldsymbol{C}(G))$ by $\boldsymbol{A}(G)$. $\boldsymbol{A}(G)$ is partially ordered with respect to the relation "$\subset$" (= set inclusion).

Let $N' \subset \mathrm{Rec}(G)$, $N' = N_1 \cup \cdots \cup N_k$, be a set of recursive symbols such that each N_i, $1 \leq i \leq k$, is p-consistent. N' is called *p-admissible* if there is some $\theta \in \boldsymbol{A}(G)$ with $\theta = \{\theta_1, \ldots, \theta_k\}$ such that $N_i \subset \theta_i$ for all $i = 1, \ldots, k$. (We have implicitly assumed that for each component of $\Gamma(G)$ there is at most one N_i contained in it. This will become clear later.)

With the above notations we prove the following.

Proposition 3.8. *N' is p-admissible iff there is a terminal derivation tree containing N' as node labels.*

Proof. Since the "only if" part is obvious, it remains to prove the "if" part.

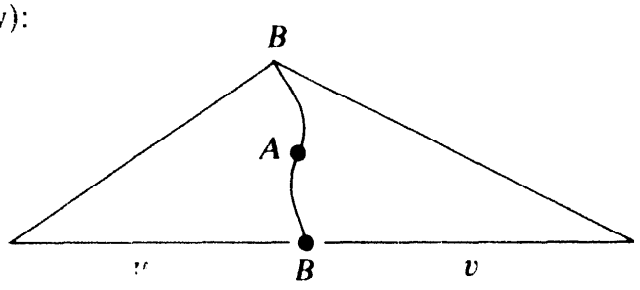
Assume that there is a terminal derivation tree Tr' containing N' as node labels. Let $N' = N_1 \cup \cdots \cup N_k$ and let $\theta_1, \ldots, \theta_k$ denote the components containing $N_1, \ldots, N_k$ respectively. We have to show that there is a terminal derivation tree Tr which contains $\theta_1 \cup \cdots \cup \theta_k$ as node labels.

Let $A \in \theta_i \backslash N_i$ be a nonterminal which does *not* occur in Tr' and let B be some nonterminal in N_i.

Since $\theta_i = [A] = [B]$, there is a directed circuit in $\Gamma(G)$ through A and B. Therefore there is a derivation

$$B \Rightarrow^* \alpha A \beta \Rightarrow^* uBv$$

in G, because G is reduced. Denote the derivation tree of this derivation by Tr_A (see diagram below):



Since B occurs in Tr', Tr_A can be inserted into Tr' at some node labeled by B. The resulting tree is obviously a terminal derivation tree and it contains A as a node label. Denote this tree by Tr''.

Applying the procedure for nonterminals in θ_i which do *not* occur in Tr'' successively, we ultimately obtain a terminal derivation tree Tr_i containing all symbols from $\theta_i \backslash N_i$ as node labels.

Repeating the whole procedure for $i = 1, \ldots, k$, we obtain a terminal derivation tree Tr containing all symbols from $\theta_1 \cup \cdots \cup \theta_k$ as node labels. Thus $\{\theta_1, \ldots, \theta_k\} \in A(G)$ and Proposition 3.8 is proved. □

In view of Proposition 3.8 we see that p-admissible sets occur in terminal derivation trees. We proceed to give the construction of constants and period systems for the commutative image of a c.f. language. We will use p-admissible sets to define periods and period systems. This provides a more detailed proof of Parikh's theorem.

The idea of the construction is as follows. Consider a large terminal derivation tree Tr and cycles in Tr. In proving Parikh's theorem one attempts to eliminate certain cycles in Tr such that after a number of eliminations a small terminal derivation tree is obtained: this tree provides the constant and the cycles provide the periods. These cycles correspond to circuits in the reachability graph. Therefore, in defining periods and period systems we will consider circuits in the reachability graph. We will see that simple circuits are sufficient to define periods and period systems, and we want to construct the periods as small as possible.

We need some notations. For any derivation tree Tr, a subpath of Tr is a subpath of some path of Tr. A subpath ω of Tr is called a *cycle* if length $(\omega) \geq 1$ and the first and last nodes of ω have the same label. ω is called a *simple cycle* if no symbol occurs twice or more on it, except that of the end nodes. Clearly, each simple cycle in Tr corresponds to some simple circuit in $\Gamma(G)$.

Construction of period systems for $\psi(L(G))$

In the following we give a construction of period systems for $\psi(L(G))$.

For a subset $U \subset \mathrm{Rec}(G)$ of recursive symbols let $\mathrm{Reach}(U)$ denote the set of all symbols reachable from U. Thus $U \subset \mathrm{Reach}(U)$.

Let $N' = N_1 \cup \cdots \cup N_k$ be a p-admissible set of nonterminals. Consider w.l.o.g. $N_1 \neq \emptyset$. Let A be an element of N_1 such that there is a simple circuit ζ in $\Gamma(G)$

through A. Let

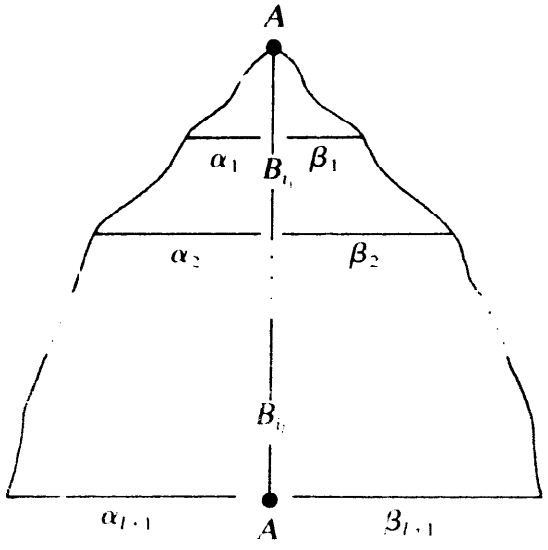
$$\zeta = (A, p_{i_1}, B_{i_1}, \ldots, B, \ldots, p_{i_l}, B_{i_l}, p_{i_{l+1}}, A),$$

where $B, B_{i_1}, \ldots, B_{i_l} \in N_1$, be this circuit. We call ζ an N_1-*circuit*. Define the derivation

$$g \equiv A \Rightarrow^{p_{i_1}} \alpha_1 B_{i_1} \beta_1 \Rightarrow^{p_{i_2}} \cdots \Rightarrow^{p_{i_{l+1}}} \alpha_{l+1} A \beta_{l+1},$$

where $\alpha_1, \beta_1, \ldots, \alpha_{l+1}, \beta_{l+1} \in \text{Reach}(N_1)^*$.

The derivation tree defined by g is illustrated by the following diagram:



Now to obtain a period we derive from the nonterminals occurring in $\alpha_{l+1}, \beta_{l+1}$ terminal words.

Let

$$\alpha_{l+1} = u_1 X_1 u_2 X_2 \ldots u_n X_n u_{n+1}, \qquad \beta_{l+1} = v_1 Y_1 v_2 Y_2 \ldots v_m Y_m v_{m+1},$$

where $X_1, \ldots, X_n, Y_1, \ldots, Y_m \in \text{Reach}(N_1) \cap N$ and $u_1, \ldots, u_{n+1}, v_1, \ldots, v_{m+1} \in T^*$.

For each nonterminal $Z \in \{X_1, \ldots, X_n, Y_1, \ldots, Y_m\}$ let

$$g_Z \equiv Z \Rightarrow^* w_Z, \quad w_Z \in T^*,$$

be a terminal derivation whose derivation tree contains *no* cycles (i.e., we want to obtain a tree which should be as small as possible).

Definine the derivation

$$h \equiv A \Rightarrow^{g} \alpha_{l+1} A \beta_{l+1} \Rightarrow^{g_{X_1}} \cdots \Rightarrow^{g_{X_n}} \Rightarrow^{g_{Y_1}} \cdots \Rightarrow^{g_{Y_m}} wA\bar{w}.$$

We call h a ζ-*derivation* and the derivation tree defined by h a ζ-*tree*. Further, we denote the set of all ζ-derivations by $\text{Der}(\zeta)$ and the set of all ζ-trees by $\text{Tree}(\zeta)$.

Fact 3.9. (i) *The depth of each* ζ*-tree is bounded by* $2\text{card}(\text{Reach}(N_1))$.

(ii) $\text{Tree}(\zeta)$ *and* $\text{Der}(\zeta)$ *are finite sets.*

Before defining the period system corresponding to N' we make a remark on simple circuits.

Consider the circuit ζ defined above:

$$\zeta = (A, p_{i_1}, B_{i_1}, \ldots, B, \ldots, p_{i_l}, B_{i_l}, p_{i_{l+1}}, A), \quad B_{i_1}, \ldots, B_{i_l} \in N_1.$$

Denote by ω_1 the path $(A, p_{i_1}, B_{i_1}, \ldots, B)$ and by ω_2 the path $(B, \ldots, p_{i_l}, B_{i_l}, A)$. Obviously,

$$\zeta' := (\omega_2, \omega_1) = (B, \ldots, p_{i_l}, B_{i_l}, p_{i_{l+1}}, A, p_{i_1}, B_{i_1}, \ldots, B)$$

is also a simple circuit in $\Gamma(G)$. The derivation h can be rearranged such that a new derivation

$$h' \equiv B \Rightarrow^* \alpha'_{l+1} B \beta'_{l+1} \Rightarrow^* w' B \bar{w}', \quad w', \bar{w}' \in T^*,$$

with $\psi(w'\bar{w}') = \psi(w\bar{w})$ can be obtained.

We say that ζ and ζ' are equivalent. For our purpose we do not distinguish between ζ and ζ'.

The idea of the construction of the period system corresponding to N' is as follows. Consider a component of N', say N_1. From the derivations corresponding to N_1-circuits we define a set of periods. The period system corresponding to N' is the union of all sets of periods obtained from the components of N'.

Now define for an N_1-circuit ζ containing $A \in N_1$

$$\Pi(\zeta) := \{\psi(ww') \mid h \equiv A \Rightarrow^* wAw' \text{ is a } \zeta\text{-derivation}\}$$

and

$$\Pi(N_1) := \bigcup_{\zeta \text{ is } N_1\text{-circuit}} \Pi(\zeta).$$

The period system corresponding to $N' = N_1 \cup \cdots \cup N_k$ is given by

$$\Pi(N') = \Pi(N_1, \ldots, N_k) := \bigcup_{i=1}^{k} \Pi(N_i).$$

Construction of constants for $\psi(L(G))$

In the following we give the construction of constants for $\psi(L(G))$. First we introduce some notations. For each simple N_i-circuit ζ a ζ-tree is also called an *N_i-tree*. An N_i-tree is also called an *$(N_1, \ldots, N_k)$-tree* (or an *N'-tree*).

The idea of the construction of constants for $\psi(L(G))$ is as follows. A terminal derivation tree Tr is called an *N'-candidate*, if every element of N' occurs in Tr as a node label. It is obvious that $(N_1, \ldots, N_k)$-trees can be inserted into an N'-candidate and the resulting tree is again a terminal one. By bounding the height of N'-candidates we get the constants for $\psi(L(G))$.

For a p-admissible set $N' \subset \mathrm{Rec}(G)$ and a derivation tree Tr define

$$\mathrm{Symb}(N') := \{A \in N \mid A \text{ occurs in some } N'\text{-candidate}\},$$

and

$$\mathrm{Symb}(\mathrm{Tr}) := \{A \in N \mid \text{occurs in Tr}\}.$$

Define the set $E(N')$ as follows. A word $w \in T^*$ is an element of $E(N')$ iff there is an N'-candidate Tr with yield w such that no element of Symb(N') occurs more than card(Symb(N')) +2 times in any path of Tr and N' is a maximal p-admissible subset of Symb(Tr). Define

$$C(N') := \{\psi(w) \mid w \in E(N')\}.$$

Proposition 3.10 (Parikh's theorem)

$$\psi(L(G)) = \bigcup_{N' \text{ p-admissible}} L(C(N'); \Pi(N')).$$

Proof. The proof of this proposition follows the pattern of the proof of Parikh's theorem. We only sketch the main point.

The inclusion "$\supset$" is straightforward. We prove "$\subset$". Let $w \in L(G)$ and Tr_w be a terminal derivation tree with frontier w.

For a derivation tree Tr let Symb(Tr) denote the set of nonterminals occurring in Tr.

Claim 1. *Among the p-admissible subsets of* Symb(Tr_w) *there is a greatest, which we denote by* N'.

Proof of Claim 1. The p-admissible subsets of Symb(Tr_w) are closed under union. □

We have Symb(Tr_w) $\subset$ Symb(N'). If no element of Symb(N') occurs more than card(Symb(N')) +2 times in any path of Tr_w, then $w \in E(N')$ and $\psi(w) \in C(N')$.

Now suppose some nonterminal $\in$ Symb(Tr_w) occurs more than card(Symb(N')) + 2 times in some path of Tr_w. Then there is a subtree Tr_{v_0} in Tr_w containing card(Symb(N')) +3 $\geq$ card(Symb(Tr_w)) +3 = s +3 nodes $v_0, v_1, \ldots, v_{s+2}$ with the same label $A \in$ Symb(Tr_w) such that $\mathrm{Tr}_{v_{j+1}}$ is a subtree of Tr_{v_j}, $0 \leq j \leq s+1$.

Obviously, there is a smallest integer $r \geq 1$ such that

$$\mathrm{Symb}(\mathrm{Tr}_{v_r}) = \mathrm{Symb}(\mathrm{Tr}_{v_{r+1}}).$$

Let $\mathrm{Tr}_{v_r, v_{r+1}}$ be the tree obtained from Tr_{v_r} by deleting the subtree $\mathrm{Tr}_{v_{r+1}}$.

Claim 2. *All cycles in* $\mathrm{Tr}_{v_r, v_{r+1}}$ *correspond to* N'*-circuits.*

Proof of Claim 2. If there is a cycle in $\mathrm{Tr}_{v_r, v_{r+1}}$ corresponding to a circuit in $\Gamma(G)$ which is not an N'-circuit, then N' would not be maximal. □

Consider the paths of $\mathrm{Tr}_{v_0, v_{r+1}}$. There are on some path two nodes ν_0, ν_1 with the following properties:

- Tr_{ν_1} is a subtree of Tr_{ν_0} and ν_0, ν_1 have the same label.
- The tree $\mathrm{Tr}_{\nu_0, \nu_1}$ obtained from Tr_{ν_0} by deleting Tr_{ν_1} is an N'-tree.

Let uAv be the frontier of $\mathrm{Tr}_{\nu_0, \nu_1}$. Then we have

$$\psi(uv) \in \Pi(N'),$$

and in Tr_w the tree $\mathrm{Tr}_{\nu_0, \nu_1}$ can be removed (cf. the proof of Proposition 3.8). Denote the resulting tree by $\mathrm{Tr}_{w'}$. Clearly, $\mathrm{Tr}_{w'}$ satisfies

(1) $\mathrm{Symb}(\mathrm{Tr}_{w'}) = \mathrm{Symb}(\mathrm{Tr}_w)$,

(2) N' is a maximal p-admissible subset of $\mathrm{Symb}(\mathrm{Tr}_{w'})$,

(3) $\mathrm{Tr}_{w'}$ is smaller than Tr_w.

Now, either $\mathrm{Tr}_{w'}$ satisfies the definition of $E(N')$ or the above deleting procedure can be applied again. Repeating this procedure for a finite number of times we ultimately obtain a derivation tree $\mathrm{Tr}_{\tilde{w}}$ satisfying (1), (2) and the definition of $E(N')$.

Thus, $\psi(w) \in L(C(N'); \Pi(N'))$ and Proposition 3.10 is proved. □

3.3. *A property of* $\psi(L(G))$

The construction in Section 3.2 gives us an effective procedure for computing a representation of $\psi(L(G))$. In this subsection we prove a property of this representation.

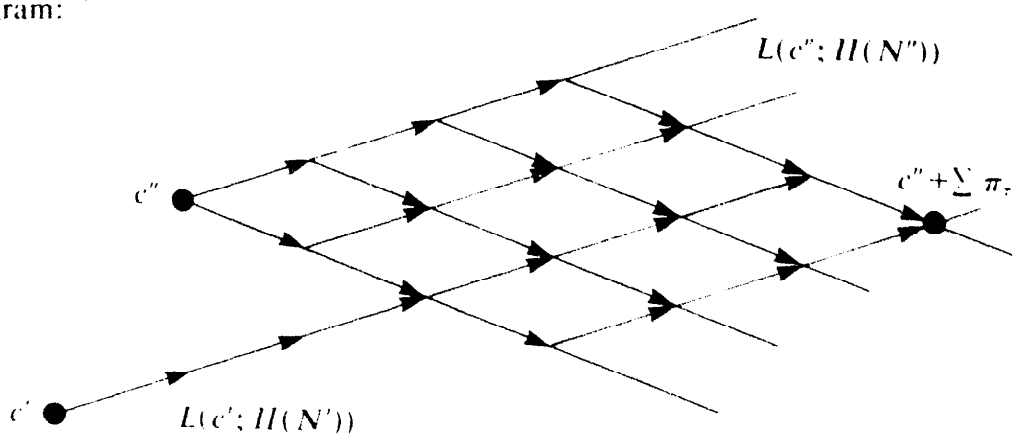
As in Proposition 3.10 let

$$\psi(L(G)) = \bigcup_{N' \text{ p-admissible}} L(C(N'); \Pi(N')). \qquad (\dagger)$$

By definition, the set $N' \subset \mathrm{Rec}(G)$ with $N' = N_1 \cup \cdots \cup N_k$, where N_i is p-consistent for all $i = 1, \ldots, k$, is p-admissible if there is some $\theta = \{\theta_1, \ldots, \theta_k\} \in A(G)$ such that $N_i \subset \theta_i$ for all $i = 1, \ldots, k$. Clearly,

$$N' \subset \bigcup_{i=1}^{k} \theta_i,$$

and $\tilde{N} := \bigcup_{i=1}^{k} \theta_i$ is p-admissible. Consider $L(C(N'); \Pi(N'))$ and $L(C(\tilde{N}); \Pi(\tilde{N}))$. We want to show that to each $c' \in C(N')$ there are certain periods in $\Pi(N'')$ such that adding these periods to c' yields an element in $L(C(N''); \Pi(N''))$, where $N'' \supset N'$ is some 'maximal' p-admissible set. This is illustrated by the following diagram:



Now, if the grammar has a one-letter terminal alphabet, then we will see that most linear sets in the representation $(\dagger)$ of $\psi(L(G))$ are 'absorbed' by those linear sets with maximal period systems, i.e., of the form $L(c''; \Pi(N''))$, $c'' \in C(N'')$. Therefore, in constructing a representation of $\psi(L(G))$ as an ultimately periodic set only those linear sets with maximal period systems have to be considered. This is the main idea in the proof of the upper bound for INEQ.

Before proving the next lemma we define

$$\mathcal{M}(G) := \left\{ \bigcup_{i=1}^{k} \theta_i \mid \{\theta_1, \ldots, \theta_k\} \in \boldsymbol{A}(G) \right\}.$$

$\mathcal{M}(G)$ is partially ordered with respect to set inclusion.

The following lemma expresses the desired property between the constants c_i's and period systems Π_j's.

Lemma 3.11. *Let $N' \neq \emptyset$, $N' \subset \mathrm{Rec}(G)$ be a p-admissible set, $N' = N_1 \cup \cdots \cup N_k$, where $N_i \neq \emptyset$ is p-consistent, $i = 1, \ldots, k$. Let $c' \in C(N')$. Then there are a p-admissible set $N'' \in \mathcal{M}(G)$ and some constant $c'' \in C(N'')$ such that*

(i) $N' \subset N''$,

(ii) *there are $\pi_1, \ldots, \pi_r \in \Pi(N'')$ such that*

$$c' + \sum_{\tau=1}^{r} \pi_\tau \in L(c''; \Pi(N'')),$$

where $r \leq \mathrm{card}(N'') - \mathrm{card}(N') \leq \mathrm{card}(N)$.

Proof. Let $\mathrm{Tr}_{w'}$ be an N'-candidate such that N' is a maximal p-admissible subset of $\mathrm{Symb}(\mathrm{Tr}_{w'})$, where $\psi(w') = c'$.

Let $\theta_1, \ldots, \theta_k$ be the components corresponding to $N_1, \ldots, N_k$ respectively. Define $\tilde{N} := \bigcup_{i=1}^{k} \theta_i$. Since $N' \subset \tilde{N}$, there are s $\tilde{N}$-circuits $\zeta_1, \ldots, \zeta_s$ with $s \leq \mathrm{card}(\tilde{N}) - \mathrm{card}(N')$, such that

$$\mathrm{Non}(\zeta_j) \cap N' \neq \emptyset, \quad \mathrm{Non}(\zeta_j) \cap \tilde{N} \neq \emptyset \quad \text{and} \quad \bigcup_{j=1}^{s} \mathrm{Non}(\zeta_j) \supset \tilde{N} \backslash N',$$

where $\mathrm{Non}(\zeta)$ denotes the set of nonterminals occurring as vertices in the circuit ζ.

From $\zeta_1, \ldots, \zeta_s$ we have s derivation trees (these are ζ_1-tree, ..., ζ_s-tree respectively), which can be inserted into $\mathrm{Tr}_{w'}$. Denote the resulting tree by $\mathrm{Tr}_{\tilde{w}}$.

Since $\bigcup_{j=1}^{s} \mathrm{Non}(\zeta_j) \supset \tilde{N} \backslash N'$, each nonterminal in $\tilde{N}$ occurs as a node label in $\mathrm{Tr}_{\tilde{w}}$. Thus $\mathrm{Tr}_{\tilde{w}}$ is an $\tilde{N}$-candidate. On the other hand, the periods $\pi_1, \ldots, \pi_s$ obtained from the derivation trees defined by $\zeta_1, \ldots, \zeta_s$ are in $\Pi(\tilde{N})$, and $\tilde{w} = c' + \sum_{\tau=1}^{s} \pi_\tau$.

Now either $\tilde{N}$ is a maximal p-admissible subset of $\mathrm{Symb}(\mathrm{Tr}_{\tilde{w}})$ or the above procedure can be repeated by considering $\tilde{N}$ as N'.

Consider the first case. Since $\mathrm{Tr}_{\tilde{w}}$ is an $\tilde{N}$-candidate, it follows from the proof of Proposition 3.10 that $\tilde{w} \in L(C(\tilde{N}); \Pi(\tilde{N}))$. Therefore, there is a $c'' \in C(\tilde{N})$ such that $\tilde{w} \in L(c''; \Pi(\tilde{N}))$. Defining $N'' := \tilde{N}$, Lemma 3.11 follows.

In the second case we repeat the above procedure for at most $\mathrm{card}(\boldsymbol{C}(G))$ times and ultimately obtain an N''-candidate $\mathrm{Tr}_{w''}$ which contains N'' as a maximal p-admissible subset of $\mathrm{Symb}(\mathrm{Tr}_{w''})$. This observation completes the proof of Lemma 3.11. □

The set N'' constructed in the proof of Lemma 3.11 is called an *N'-superset.*

4. Representation of c.f. 1-letter alphabet languages as ultimately periodic sets

In this section we present a technique for proving the upper bound of INEQ. The main problem is to achieve the following fact: For two c.f. grammars with the same terminal alphabet $T=\{0\}$ let $\triangle$ denote the symmetric difference $[L(G_1)\backslash L(G_2)]\cup[L(G_2)\backslash L(G_1)]$. We shall show that

$$\triangle \neq \emptyset \quad \text{iff there is some } n\in\mathbb{N}_0 \text{ with } n\leqslant 2^{Q(\#(G_1,G_2))} \text{ such that } 0^n\in\triangle,$$

where Q is a fixed polynomial and $\#(G_1, G_2)$ denotes the size of the input grammars G_1, G_2.

The idea is to represent $L(G_i)$ as an ultimately periodic set. By using the results of Section 3 the above fact can be proved.

4.1. Ultimately periodic sets

Ultimately periodic sets were used in [7] for proving an elementary recursive upper bound for the equivalence problem for extended regular expressions over a 1-letter alphabet.

Definition 4.1. A subset U of $\mathbb{N}_0$ is said to be *ultimately periodic* (u.p. for short), if it can be represented in the form

$$U = F\cup L(C;p),$$

where $F, C\subset\mathbb{N}_0$ are finite sets, $p\in\mathbb{N}_0$ and $F\cap L(C;p)=\emptyset$.

We call F the finite part of U, an element $c\in C$ a constant of U and p the period of U. $(F;C;p)$ is called a *representation of U*.

In the following, let $G=(N,\{0\},S,P)$ be a reduced c.f. grammar. (We shall use the notions introduced in Section 3.)

In order to obtain a 'small' representation for $\psi(L(G))$ as an u.p. set we need the following lemma.

Lemma 4.2. *Let $\Pi=\{\pi_1,\ldots,\pi_t\}$ be a subset of $\mathbb{N}_0$ with $\mathrm{Max}(\Pi)=l$. Then $L(0;\Pi)$ has a representation of the form*

$$L(0;\Pi)=F\cup L(c;p)$$

as an u.p. set such that

(1) $p=\gcd(\Pi)$ (gcd = *greatest common divisor*),

(2) $c=(\sum_{i=1}^{t}\pi_i)^2\leqslant(tl)^2$ *and* $\mathrm{Max}(F)<c$.

Proof. For the proof, cf. [7, proof of Lemma 2, p. 25]. □

4.2. 'Small' representation of $\psi(L(G))$ as a u.p. set

In the following we show that $\psi(L(G))$ has a 'small' representation as an u.p. set. Consider

$$\psi(L(G)) = \bigcup_{N' \text{ p-admissible}} L(C(N'); \Pi(N')). \qquad (\star)$$

We can use the technique of Rangel [7] to obtain a representation for $\psi(L(G))$ as an u.p. set. Such a direct application does not provide us with a 'small' representation. Also the refined technique in [1] is not applicable. Indeed, without using the results of Section 3 we get in both cases a double-exponential upper bound for the integer n stated in the fact we want to prove.

Lemma 4.3. *For each element $v \in \bigcup_{N'} C(N') \cup \Pi(N')$ it holds that*

$$v \leq 2^{Q_1(\#G)},$$

where Q_1 is a fixed polynomial.

Proof. *Case* 1. $v = \pi \in \cup \Pi(N')$. There is a ζ-tree for w_π such that $\psi(w_\pi) = \pi$. Since in this tree there is only one simple cycle, Lemma 4.3 holds.

Case 2. $v = c \in \cup C(N')$. There is per definition of $C(N')$ a terminal derivation tree for w_c, $\psi(w_c) = c$, such that each nonterminal does not appear more than $\text{card}(N)+2$ times in any path of this tree. Thus Lemma 4.3 holds. □

Lemma 4.4. *Let $L = L(c; \Pi)$ be a linear set occurring in the representation ($\star$) of $\psi(L(G))$. Then there is a constant μ such that L has the representation*

$$L(c; \Pi) = F \cup L(\bar{c}; p)$$

as an u.p. set such that

(1) $p = \gcd(\Pi)$

(2) $\bar{c} \leq \mu \leq 2^{Q_2(\#G)}$ *and* $\text{Max}(F) < \bar{c}$

for some fixed polynomial Q_2.

Proof. The proof follows by applying Lemmas 4.2 and 4.3. □

We now prove the main result of this section.

Proposition 4.5. *Let $\theta_1, \ldots, \theta_s$ be the components of $\Gamma(G)$. Further, let*

$$p_G := \gcd(\Pi(\theta_1)) \cdot \ldots \cdot \gcd(\Pi(\theta_s)).$$

Then $\psi(L(G))$ has the representation

$$\psi(L(G)) = F_G \cup L(C_G; p_G) \qquad (*)$$

as an u.p. set such that

$$\mathrm{Max}(F_G) < \mathrm{Min}(C_G) \leq \mathrm{Max}(C_G) \leq 2^{Q_3(\neq G)}$$

for some fixed polynomial Q_3.

Further, it holds that

$$p_G \leq 2^{Q_4(\neq G)}$$

for some fixed polynomial Q_4.

Proof. We first show the last statement. Observe that

$$\gcd(\Pi(\theta_i)) \leq \mathrm{Max}(\Pi(\theta_i)) \leq 2^{Q_1(\neq G)}.$$

On the other hand, $s \leq \mathrm{card}(N)$. This proves the last statement.

Before constructing the representation (*) we make some remarks. Let $N' \neq \emptyset$ be a p-admissible set and $L(c'; \Pi(N'))$ be a linear set occurring in the representation (☆), where $c' \in C(N')$. Let $\theta := \{\theta_{i_1}, \ldots, \theta_{i_k}\} \in A(G)$ such that $N'' := \bigcup \theta_{i_j}$ is an N'-superset.

By Lemma 3.11 we see that the linear set $L(c'; \Pi(N'))$ is 'absorbed' by the linear set $L(c''; \Pi(N''))$ for some $c'' \in C(N'')$. We state this fact more precisely in the following claim.

Claim. There is a constant $\bar{\mu} = \bar{\mu}(G)$ *such that for arbitrary* N', N'', c' *and* c'' *as above it holds that*

$$\forall v \in L(c'; \Pi(N')) \; : \; v > \bar{\mu} \Rightarrow v \in L(c''; \Pi(N'')).$$

Further,

$$\bar{\mu} \leq 2^{Q_5(\neq G)}$$

for some fixed polynomial Q_5.

Proof of Claim. From Lemma 3.11 it follows that there are $c'' \in C(N'')$ and $\pi_1, \ldots, \pi_r \in \Pi(N'')$, $r \leq \mathrm{card}(N'')$, such that

$$c' + \sum_{j=1}^{r} \pi_j = c'' + \sum_{l=1}^{t} \lambda_l \bar{\pi}_l, \quad \lambda_1, \ldots, \lambda_t \in \mathbb{N}_0, \; \bar{\pi}_1, \ldots, \bar{\pi}_t \in \Pi(N'').$$

Hence,

$$\gcd(\pi_1, \ldots, \pi_r, \bar{\pi}_1, \ldots, \bar{\pi}_t) \,|\, |c' - c''|,$$

and therefore

$$\gcd(\Pi(N'')) \,|\, |c' - c''|.$$

Consider the construction of the constant $\bar{c}$ in the representation $F \cup L(\bar{c}; p)$ of $L(c; \Pi)$ as an u.p. set by Lemma 4.4. Let $\bar{c}'$, $\bar{c}''$ be the constants in the representations of $L(c'; \Pi(N'))$ and $L(c''; \Pi(N''))$ as u.p. sets. Since $\Pi(N'') \supset \Pi(N')$ implies

$$\gcd(\Pi(N'')) \,|\, \gcd(\Pi(N')) \quad \text{and} \quad \gcd(\Pi(N'')) \,|\, |c' - c''|,$$

we see that

$$\forall v \in L(c'; \Pi(N')) \ : \ v > \bar{c}'' \Rightarrow v \in L(c''; \Pi(N'')).$$

Therefore, choosing

$$\bar{\mu} = \bar{\mu}(G) := \mathrm{Max}\{\bar{c} \mid \bar{c} \text{ is the constant in the representation } F \cup L(\bar{c}; p) \text{ of a linear set } L(c; \Pi) \text{ in } (\star) \text{ as a u.p. set by Lemma 4.4}\}$$

we obtain the first statement of the claim.

Because $\bar{\mu} \leqslant 2^{Q_3(\#G)}$, the second statement of the claim follows. Thus the proof of the claim is complete. □

We return to the proof of the lemma. For each $N'' \in \mathcal{M}(G)$ define

$$p_{N''} := \gcd(\Pi(N''))$$

and

$$D(N'') := \{v \mid v \in L(C(N''); \Pi(N'')) \text{ and } \bar{\mu} \leqslant v \leqslant \bar{\mu} + p_{N''}\}.$$

Then we have

$$\psi(L(G)) = F_G \cup \bigcup_{N'' \in \mathcal{M}(G)} L(D(N''); p_{N''}),$$

where

$$F_G := \psi(L(G)) \setminus \left[\bigcup_{N'' \in \mathcal{M}(G)} L(D(N''); p_{N''}) \right].$$

This fact follows from the above claim.

On the other hand, we have

$$\forall \theta = \{\theta_{i_1}, \ldots, \theta_{i_k}\} \in \mathcal{A}(G): \quad p_{N''} \mid \gcd(\Pi(\theta_{i_j})), \ j = 1, \ldots, k,$$

where $N'' = \bigcup_{j=1}^{k} \theta_{i_j}$, since $\Pi(\theta_{i_j}) \subset \Pi(N'')$.

Therefore, $p_{N''} \mid p_G$ for all $N'' \in \mathcal{M}(G)$.

Defining

$$C_g := \left\{ v \in \bigcup_{N'' \in \mathcal{M}(G)} L(D(N''); p_{N''}) \mid v \leqslant \bar{\mu} + p_G \right\},$$

the first statement of Proposition 4.5 follows.

The fact that

$$\mathrm{Max}(F_G) < \mathrm{Min}(C_G) \leqslant \mathrm{Max}(C_G) \leqslant 2^{Q_3(\#G)}$$

for some fixed polynomial Q_3 can be easily verified. This completes the proof of Proposition 4.5. □

From Proposition 4.5 we have the following proposition.

Proposition 4.6. *Let G_1 and G_2 be two c.f. grammars with the same terminal alphabet $T = \{0\}$. Then*

$$\Delta := [L(G_1) \setminus L(G_2)] \cup [L(G_2) \setminus L(G_1)] \neq \emptyset$$

iff there is some nonnegative integer $n \in \mathbb{N}_0$ with $n \leq 2^{Q(\#(G_1,G_2))}$ such that $0^n \in \Delta$, where Q is a fixed polynomial.

Proof. Proposition 4.6 immediately follows from Proposition 4.5. □

5. The complexity of the inequivalence problem

In this section we classify the complexity of INEQ. From Proposition 4.6 we prove the following.

Proposition 5.1. INEQ *is in* Σ_2^p.

Proof. From Proposition 4.6 we have $L(G_1) \neq L(G_2)$ iff $\Delta \neq \emptyset$ iff there is some $n \in \mathbb{N}_0$ such that $0^n \in \Delta$ and $n \leq 2^{Q(\#(G_1,G_2))}$, where Q is some fixed polynomial.

An alternating Turing machine M with at most one alteration operating in polynomial-time, starting with an existential state recognizes INEQ as follows: Guess a binary representation of n and determine whether $0^n \in \Delta$. Thus INEQ $\in \Sigma_2^p$. □

We now show that INEQ is log-hard for Σ_2^p.

Proposition 5.2. INEQ *is log-hard for* Σ_2^p.

Proof. If we can construct a log-space reduction from the inequivalence problem for integer expressions denoted by N-INEQ to INEQ, then Proposition 5.2 is proved, since N-INEQ is known to be log-complete for Σ_2^p (cf. [9]).

Integer expressions are expressions involving nonnegative integers written in binary representation without leading zeros. + and $\cup$ are the binary operations. Integer expressions define subsets of $\mathbb{N}_0$.

The inequivalence problem for integer expressions is defined as follows: Given two integer expressions γ_1, γ_2, determine whether they define different subsets of $\mathbb{N}_0$.

Integer expressions can be simulated by c.f. 1LTA grammars as follows. For each expression γ we construct a c.f. 1LTA grammar $G(\gamma) = (N, \{0\}, S, P)$ such that

$$n \text{ is in the set defined by } \gamma \Leftrightarrow 0^n \in L(G(\gamma)).$$

Since + corresponds to concatenation and $\cup$ corresponds to union of c.f. languages, these operations can be simulated by c.f. productions. The only problem is to describe integers in binary representation without leading zeros by not 'too many' c.f. productions. This can be done as in the proof of Proposition 2.2.

Thus a log-space reduction from N-INEQ to INEQ can be constructed. This completes the proof of Proposition 5.2. □

From Propositions 5.1 and 5.2 we obtain the following theorem.

Theorem 5.3. INEQ *is log-complete for* Σ_2^p.

Since the grammar constructed in the proof of Proposition 5.2 generates a finite language, we also get the following.

Corollary 5.4. *The inequivalence problem for c.f. 1LTA grammars generating finite languages is log-complete for* Σ_2^p.

6. Concluding remarks

In the previous sections we have characterized the complexity of MEMBER and INEQ. The grammars we considered have a 1-letter terminal alphabet. As mentioned in the Introduction, the equivalence and the commutative equivalence problems are the same. Hence, we can consider these grammars as commutative grammars as in [5]. Thus we can work over a free commutative monoid instead of a free monoid. And words are commutative words in this case (c.f. the proof of Proposition 2.1); they are coded by their exponent sums in binary representation. It is not hard to see that all the results in this paper also hold for the commutative case.

Consider on the other hand the inequivalence problem INEQ($\{0\}, \{\cup, \cdot, ^2, ^*\}$), i.e., the inequivalence problem for regular expressions over the 1-letter alphabet $\{0\}$ with the operations $\cup, \cdot, ^2, ^*$. Clearly, such regular expressions can be simulated by c.f. 1LTA grammars as in the proof of Proposition 5.2. Thus INEQ($\{0\}, \{\cup, \cdot, ^2, ^*\}$) is in Σ_2^p. Furthermore, integer expressions can also be simulated by such regular expressions. This implies that INEQ($\{0\}, \{\cup, \cdot, ^2, ^*\}$) is log-hard for Σ_2^p and hence log-complete for Σ_2^p.

Acknowledgment

The author wishes to thank B. Becker and H.U. Simon for some critical remarks concerning this paper, and an anonymous referee for many helpful suggestions in clarifying the presentation of this work.

References

[1] M. Fürer, The complexity of the inequivalence problem for regular expressions with intersection, *Proc. 7th Colloq. on Automata, Languages and Programming*, Lecture Notes in Computer Science **85** (Springer, Berlin, 1980) pp. 234–245.

[2] S. Ginsburg, *The Mathematical Theory of Context-Free Languages* (McGraw-Hill, New York, 1966).
[3] H.B. Hunt III, D.J. Rosenkrantz and T.G. Szymanski, On the equivalence, containment, and covering problems for the regular and context-free languages, *J. Comput. System Sci.* **12** (1976) 222–268.
[4] D.T. Huynh, The complexity of semilinear sets, *Elektr. Inform.-verarbeitung & Kybern.* **6** (1982) 291–338.
[5] D.T. Huynh, Commutative grammars: The complexity of uniform word problems, *Inform. & Control* **57** (1984) 21–39.
[6] D.T. Huynh, Remarks on the complexity of an invariant of context-free grammars, *Acta Informatica* **7** (1982) 89–99.
[7] J.L. Rangel, The equivalence problem for regular expressions over one letter is elementary, *15th Ann. Symp. on Switching and Automata Theory* (1974) pp. 24–27.
[8] L.J. Stockmeyer, The complexity of decision problems in automata theory and logic, Rept. TR-133, M.I.T., Project MAC, Cambridge, MA, 1974.
[9] L.J. Stockmeyer and A.R. Meyer, Word problems requiring exponential time: Preliminary report, *Proc. 5th Ann. ACM Symp. on the Theory of Computing* (1973) pp. 1–9.