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

by Thiet-Dung HUYNH [*]

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

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 in [3] by Hunt III, Rosenkrantz and Szymanski.

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 sciences. 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 inequivalence problem, concerning regular expressions, various classes of grammars have been studied first by Meyer, Stockmeyer, Hunt III, Rosenkrantz Szymanski ...

In this paper we are also concerned with these decision problems in connection with context-free grammars with 1-letter terminal alphabet. In this case there is no difference between the equivalence problem and the commutative equivalence problem, which has been studied by the author in [5] and [6]. (The reason of our study on the complexity of commutative equivalence problems is that commutativity provides us necessary conditions whose complexity 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 second level of the polynomial-time hierarchy introduced in [9] by Meyer and Stockmeyer.

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

This paper consists of 6 sections. In section 1 we review known definitions used in this work. 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

[*] Current address : Department of Mathematics, University of Chicago, Chicago, Illinois 60637, U.S.A.

0272-5428/82/0000/0021$00.75 © 1982 IEEE

expressed as ultimately periodic sets by "small" representations (cf. section 4). This later fact will be proved in section 4. In section 5 we classify the complexity of the inequivalence problem. We close 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. ε denotes the empty word. A subset $L \subset \Sigma^*$ is called a language. A language $L \subset \Sigma^*$ with $\mathrm{card}(\Sigma) = 1$ is called a 1-letter alphabet la nguage (or 1LA language for short), where card(S) denotes the cardinality of 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 $P \subset N\mathrm{x}(N \cup T)^*$ is the finite set of productions. The language generated by G is denoted by L(G). The relations $\xrightarrow[G]{}$, $\xrightarrow[G]{*}$, $\xrightarrow[G]{+}$ are defined as usual. We often write $\rightarrow$, $\xrightarrow{*}$, $\xrightarrow{+}$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 commutatively 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 give below the definition of "semilinear sets" for the sake of completeness. In the following $\mathbb{N}_o$ denotes the set of nonnegative integers and $\mathbb{Z}$ the set of integers.

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

$$L(C;\Pi) := \{ c + \sum_{\pi\in\Pi} \lambda_\pi \pi \mid c \in C,\ \lambda_\pi \in \mathbb{N}_o \}.$$

If $C = \{c_1,\dots,c_r\}$ and $\Pi = \{\pi_1,\dots,\pi_s\}$, we also write $L(c_1,\dots,c_r;\pi_1,\dots,\pi_s)$. Further, if $C = \emptyset$, then $L(C;\Pi) := \emptyset$.

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

If $L = L(c;\Pi)$ is a lin. set, so we call $(c;\Pi)$ a representation of L. Obviously, a lin. set may have different representations. If $SL = L(c_1;\Pi_1) \cup \dots \cup L(c_m;\Pi_m)$ is a s.l. set, then $(c_1;\Pi_1) \cup \dots \cup (c_m;\Pi_m)$ is called a representation of SL. Two representations of s.l. sets are called equivalent, if they define the same s.l. set.

For a finite alphabet $U = \{u_1,\dots,u_k\}$ define the following mapping

$$\psi_U : U^* \longrightarrow \mathbb{N}_o^k$$
$$w \longmapsto \psi_U(w) := (|w|_{u_1}, |w|_{u_2}, \dots, |w|_{u_k}),$$

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

We shall characterize the complexity of problems in terms of known complexity classes. We assume that the reader is familiar with 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 $\Sigma_o^p \subset \Sigma_1^p \subset \Sigma_2^p \subset \ldots \subset \Sigma_k^p \subset \ldots$,

$\Pi_o^p \subset \Pi_1^p \subset \Pi_2^p \subset \ldots \subset \Pi_k^p \subset \ldots$,

where $\Sigma_o^p = \Pi_o^p = P$ 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 and a nonnegative integer $n \in \mathbb{N}_o$ it is to determine whether $0^n \in L(G)$, where $G = (N,\{0\},S,P)$ is the given grammar.

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 it is to determine whether $L(G_1) \neq L(G_2)$.

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 complete for Σ_2^p is the inequivalence problem for integer expressions. This is proved by Meyer and Stockmeyer in [9]. Recently, it has been proved in [4] that the inequivalence problem for s.l. sets is also complete for this class of the polynomial-time hierarchy.

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 produces 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 study 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 "small" representation of a c.f.1LTA language as ultimately periodic set does exist. And this provides us the desired upper bound for INEQ.

§2. The Complexity of MEMBER.

Proposition 2.1. MEMBER is in NP.

Proof. Since it has been proved in [5] that the membership problem for c.f. commutative grammars is in NP (in [5] it is called the uniform word problem), it follows that MEMBER is in NP, too. We reproduce here only the main idea. For further details the reader is referred to [5].

We use some notations. For a finite alphabet U let $U^{\oplus}$ denote the free commutative monoid generated by U. Let $G=(N,T,S,P)$ be a c.f. grammar. D(G) denotes the set of terminal derivation words of G. A subset P' of P is called connected, if there is a terminal derivation tree such that the set of productions occuring in it is exactly P'. Further for any $u \in P^{\oplus}$ let P(u) denote the set of elements of P occuring in u. If a terminal derivation tree contains exactly P' as productions and if each production occurs at most card(P)+2 times, it is called P'-minimal.

Claim 1. For each $u \in P^{\oplus}$, $u \in \psi(D(G))$*,iff the following conditions hold.

(1) P(u) is a connected subset of P and there is a P(u)-minimal terminal derivation

* It is equivalent to consider u as an element of $P^{\oplus}$ or as the integer vector defined by the exponent sums.

tree Tr_u* corresponding to a derivation word u^* such that $\psi(u^*) \le u$.

(2) $\kappa(u) = \psi(w) - \psi(S)$, where $w \in T^*$ and κ is the homomorphism

$$\kappa : P^{\oplus} \longrightarrow \mathbb{Z}^1$$
$$(A \to \alpha) \longmapsto (\psi(\alpha) - \psi(A))$$

and $\psi : (N \cup T)^* \to \mathbb{N}_0^1$ is the Parikh-mapping on $N \cup T$.

Claim 2. $\psi(w) = (e_1,\dots,e_s) \in \psi(L(G))$, iff there is some $u = (f_1,\dots,f_r) \in \psi(D(G))$ such that

$\kappa(u) = \psi(w) - \psi(S)$ and $\Sigma f_j \le e.d^{\#G}$

for some fixed constant d, where $e = \Sigma e_i$ and $\#G$ denotes the size of G.

From claim 1 and claim 2 proposition 2.1 follows. □

Proposition 2.2. MEMBER is log-hard for NP.

Proof. Construct a log-space reduction from the knapsack problem to MEMBER. We obmit the details. □

From propositions 2.1 and 2.2 we obtain

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

Corollary 2.4. The membership problem for c.f.1LTA 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 exactly the s.l. sets, as stated in Parikh 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. We assume w.l.o.g. that G is reduced.

Let $L_1 \cup \dots \cup L_m$, $L_i = L(c_i;\Pi_i)$, $i = 1,\dots,m$ be 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. This property can be achieved by a detailed analysis of the proof of Parikh's theorem. More precisely, we shall prove that for certain constant c_i there is some j and a subset $\Pi \subset \Pi_j$ such that $c_i + \sum_{\pi\in\Pi} \pi \in L(c_j;\Pi_j)$.

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_j's.

3.1. 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 bipatite digraph $\Gamma(G) = (W,F)$ satisfying the following conditions.

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

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

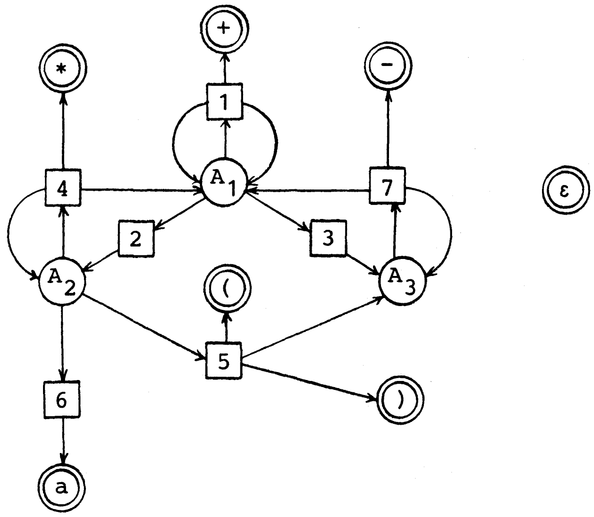
- $p = (A \to \varepsilon)$ is an ε-production: (A,p), (p,ε) are edges in F.
- $p = (A \to X_1 \dots X_n)$, $n \ge 1$: the edges (A,p), $(p,X_1),\dots,(p,X_n)$ are edges in F.

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

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

which are denoted by $p_1,\dots,p_7$, respectively, is the following bipartite digraph, where a vertex p_i from P is denoted by $\boxed{i}$, a vertex from N by $\textcircled{A_j}$ and a vertex from $T \cup \{\varepsilon\}$ by $\textcircled{\textcircled{a}}$. (See figure below.)

Remark 3.2. From the definition of $\Gamma(G)$



it is easy to see that all the nonterminals of G_1 are reachable, since it holds that $A \overset{+}{\Rightarrow} \alpha B\beta$, $A,B \in N$, $\alpha,\beta \in V^*$, iff there is a directed path in $\Gamma(G)$ leading from A to B.

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 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 consider only equivalence classes which contain at least two vertices from $N \cup P$. It is straight forward 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* for short.

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

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

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

Definition 3.5. Let $N' \subset \mathrm{Rec}(G)$ be a subset of recursive symbols. N' is called *p-consistent*, if the following conditions hold

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

(2) For every vertex $B \in N' \subset [A]$ there is a directed circuit $(B, p_{j_1}, B_{j_1}, \dots, B_{j_l}, p_{j_{l+1}}, B)$ with $B_{j_1}, \dots, B_{j_l} \in N'$.

Remark 3.6. From definition 3.5 it follows that for each component [A] of $\Gamma(G)$ the set 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.)

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 provides 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 node label in Tr.

Let $\underline{C}(G)$ denote the set of components of $\Gamma(G)$ and $\mathcal{P}(\underline{C}(G))$ denote the power set of $\underline{C}(G)$. An element $\Theta \in \mathcal{P}(\underline{C}(G))$ 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 of $\mathcal{P}(\underline{C}(G))$ by $\underline{A}(G)$. $\underline{A}(G)$ is partially ordered with respect to "$\subset$".

Let $N' \subset \mathrm{Rec}(G)$, $N' = N_1 \cup \ldots \cup N_k$, 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 $\Theta \in \underline{A}(G)$ with $\Theta = \{\Theta_1, \ldots, \Theta_k\}$ such that $N_i \subset \Theta_i$ for all $i=1,\ldots,k$. (We have implicitly assume that for each component of $\Gamma(G)$ there is at most one N_i contained in it. This will be clear later.)

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

Proof. We obmit the proof. □

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 $\mathrm{length}(\omega) \geq 1$ 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 cycle in Tr corresponds to some circuit in $\Gamma(G)$ and each 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 Reach(U) denote the set of all symbols reachable from U. Thus $U \subset \mathrm{Reach}(U)$.

Let $N' = N_1 \cup \ldots \cup N_k$ 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 $\underline{N_1\text{-circuit}}$. Define the derivation

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

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

$$\alpha_{l+1} = u_1 X_1 \ldots u_n X_n u_{n+1} ,$$

and

$$\beta_{l+1} = v_1 Y_1 \ldots v_m Y_m v_{m+1} ,$$

where $X_1, \ldots, X_n, Y_1, \ldots, Y_m \in \mathrm{Reach}(N_1) \cap N$ 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

$$g_Z \equiv Z \overset{*}{\Rightarrow} w_Z , \ w_Z \in T^*,$$

be a terminal derivation whose derivation tree contains no cycles. Define

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

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 $\mathrm{Der}(\zeta)$ and the set of all ζ-trees by $\mathrm{Tree}(\zeta)$.

Fact 3.9.(1) The depth of each ζ-tree is bounded by $2\mathrm{card}(\mathrm{Reach}(N_1))$.

(2) $\mathrm{Tree}(\zeta)$ and $\mathrm{Der}(\zeta)$ are finite sets.

Before defining the period system corresponding to N_1 we make a remark on simple circuits. Consider the circuit ζ defined above

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

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, $\zeta' := (\omega_2, \omega_1)$ is also a simple circuit in $\Gamma(G)$. The derivation h can be rearranged such that a new derivation

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

with $\psi(w'\bar{w}') = \psi(w\bar{w})$ can be obtained. We say that ζ and ζ' are equivalent and for our purpose we do not distinguish ζ and ζ'.

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

$$\Pi(\zeta) := \{\psi(ww') \mid h = A \overset{*}{\Rightarrow} wAw' \in \mathrm{Der}(\zeta)\}$$

and

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

The period system corresponding to $N' = N_1 \cup \ldots \cup N_k$ is

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

Construction of Constants for $\psi(L(G))$.

In the following we give the construction of constants for $\psi(L(G))$. 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 *$(N_1,\dots,N_k)$-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 node label. It is obvious that $(N_1,\dots,N_k)$-trees can be inserted into an N'-candidate and the resulting tree is a terminal one. By bounding the height of N'-candidates we obtain the constants for $\psi(L(G))$.

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

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

and

$\mathrm{Symb}(\mathrm{Tr}) := \{ A \in N \mid A$ 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 such that no element of $\mathrm{Symb}(N')$ occurs more than $\mathrm{card}(\mathrm{Symb}(N'))+2$ times in any path of Tr and N' is a maximal p-admissible subset of $\mathrm{Symb}(\mathrm{Tr})$. 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 theorem follows the pattern of the original 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.

Claim 1. Among the p-admissible subsets of $\mathrm{Symb}(\mathrm{Tr}_w)$ there is a greatest one which is denoted by N'.

Proof of claim 1. Since p-admissible subsets are closed under union, claim 1 follows. □

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

Now suppose some nonterminal $\in \mathrm{Symb}(\mathrm{Tr}_w)$ occurs more than $\mathrm{card}(\mathrm{Symb}(N'))+2$ times in some path of Tr_w. Then there is a subtree Tr_{v_o} in Tr_w containing $\mathrm{card}(\mathrm{Symb}(N'))+3 \geq \mathrm{card}(\mathrm{Symb}(\mathrm{Tr}_w))+3 = s+3$ nodes $v_o, v_1, \dots, v_{s+2}$ with the same label $A \in \mathrm{Symb}(\mathrm{Tr}_w)$ 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 $r \geq 1$ with

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

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. $\mathrm{Tr}_{v_r,v_{r+1}}$ contains cycles and 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 no N'-circuit, then N' would not be maximal. □

Consider the paths of $\mathrm{Tr}_{v_r,v_{r+1}}$. There is on some path two nodes ν_o, ν_1 with the following properties

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

We have

$$\psi(\mathrm{frontier}(\mathrm{Tr}_{\nu_o,\nu_1})) \in \Pi(N'),$$

and in Tr_w the tree $\mathrm{Tr}_{\nu_o,\nu_1}$ can be removed (cf. 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 $\mathrm{Tr}_{\bar{w}}$ 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 subsection 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 be

$$\psi(L(G)) = \bigcup_{N'\ \text{p-admissible}} L(C(N');\Pi(N')) \quad (1)$$

Let $\mathfrak{M}(G)$ denote the set of p-admissible subsets of the form $\Theta_1 \cup .. \cup \Theta_k, \{\Theta_1,\dots,\Theta_k\} \in \underline{A}(G)$. $\mathfrak{M}(G)$ is partially ordered with respect to set inclusion. $\emptyset$ is the least element. The number of elements of $\mathfrak{M}(G)$ may be exponential in terms of the number of components of $\Gamma(G)$. The following lemma expresses the desired property between the constants c_i's and period systems Π_j's.

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

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

(2) There is a subset $\Pi = \{\pi_1,\dots,\pi_r\} \subset \Pi(N'')$ such that $c' + \sum_{\tau=1}^{r} \pi_\tau \in L(c'';\Pi(N''))$, where $r \leq \text{card}(N'') - \text{card}(N') \leq \text{card}(N)$.

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

Let $\Theta_1,\dots,\Theta_k$ be the components corresponding to $N_1,\dots,N_k$ respectively. Define $\tilde{N} := \bigcup_{i=1}^{k} \Theta_i$. Since $N' \subset \tilde{N}$, there are s $\tilde{N}$-circuits $\zeta_1,\dots,\zeta_s$, $s \leq \text{card}(\tilde{N}) - \text{card}(N')$, such that

$$Non(\zeta_j) \cap N' \neq \emptyset \ , \ Non(\zeta_j) \cap N'' \neq \emptyset \ ,$$

and
$$\bigcup_{j=1}^{s} Non(\zeta_j) \supset \tilde{N} \setminus N' ,$$

where $Non(\zeta)$ denotes the set of nonterminals occuring as vertices in the circuit ζ.

From $\zeta_1,\dots,\zeta_s$ we have s derivation trees (these are ζ_1-tree,..., ζ_s-tree respectively) , which can be inserted into $Tr_{w'}$. Denote the resulting tree by $Tr_{\tilde{w}}$. Clearly, $Tr_{\tilde{w}}$ is an $\tilde{N}$-candidate.

Now either $\tilde{N}$ is a maximal p-admissible subset of $Symb(Tr_{\tilde{w}})$ or the above procedure can be repeated by considering $\tilde{N}$ as N'. In the first case let $N'' := \tilde{N}$. Then there is some $c'' \in C(N'')$ such that $\psi(w'') \in L(c'';\Pi(N''))$ (cf. proof of proposition 3.10), and we are finished. In the second case we repeat the above procedure for at most $\text{card}(\underline{C}(G))$ times and ultimately obtain an N''-candidate $Tr_{w''}$ which contains N'' as a maximal p-admissible subset of $Symb(Tr_{w''})$. 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.1LA Languages as Ultimately Periodic Sets.

In this section we present a technic 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 Δ denote the symmetric difference $[L(G_1) \setminus L(G_2)] \cup [L(G_2) \setminus L(G_1)]$. We show that

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

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

The idea is to represent $L(G_i)$, $i=1,2$, as ultimately periodic set. By using the results of §3 the above fact can be proved.

4.1. Ultimately Periodic Sets.

Ultimately periodic sets were used in [7] in proving an elementary recursive upper bound for the equivalence problem for extended regular expressions over an 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 $G=(N,\{O\},S,P)$ be a reduced c.f.grammar. (We shall use the notions introduced in §3.) In order to obtain "small" representation for $\psi(L(G))$ as u.p. set we need

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

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

as u.p. set such that

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

(2) $c = (\sum_{j=1}^{t} \pi_j)^2 \le (tl)^2$ and $\mathrm{Max}(F) < c$.

Proof. (cf.[7], proof of lemma 2.) □

4.2. "Small" Representation of $\psi(L(G))$ as U.P. Set.

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

Lemma 4.3. For each element v as constant or period of $\psi(L(G))$ it holds that

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

where Q_1 is a fixed polynomial.

Proof. Case 1. v is a period. There is a ζ-tree for w_v, $\psi(w_v) = v$. From part(1) of fact 3.9 the depth of each ζ-tree is linearly bounded, lemma 4.3 follows.

Case 2. v is a constant. There is per definition of $C(N')$ a terminal derivation tree for $w_v, \psi(w_v) = v$, such that each nonterminal does not appear more than 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 lin. set occuring in the representation (1) 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 u.p.set such that

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

(2) $\bar{c} \le \mu \le 2^{Q_2(\#G)}$ and $\mathrm{Max}(F) < \bar{c}$,

for some fixed polynomial Q_2.

Proof. Applying lemmas 4.2 and 4.3. □

We prove the main result of this section

Proposition 4.5. Let $\Theta_1,\dots,\Theta_s$ be the components of $\Gamma(G)$. Further let

$$p_G := \gcd(\Pi(\Theta_1))\dots\gcd(\Pi(\Theta_s)) .$$

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

$$\psi(L(G)) = F_G \cup L(C_G;p_G) \qquad (2)$$

as u.p. set such that

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

for some fixed polynomial Q_3. Further it holds that

$$p_G \le 2^{Q_4(\#G)}$$

for some fixed polynomial Q_4.

Proof. We first show the last statement. Observe that

$$\gcd(\Pi(\Theta_i)) \le \mathrm{Max}(\Pi(\Theta_i)) \le 2^{Q_1(\#G)} .$$

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

Before constructing the representation (2) we make some remarks. Let $N' \ne \emptyset$ be a p-admissible set and $L(c';\Pi(N'))$ be a lin. set occuring in the representation (1), where $c' \in C(N')$. Let $\Theta := \{\Theta_{i_1},\dots,\Theta_{i_k}\} \in \underline{A}(G)$ such that $N" := \cup\,\Theta_{i_j}$ is an N'-superset.

Claim. There is a constant $\bar{\mu} = \bar{\mu}(G)$ and an element $c" \in C(N")$ such that for every $v \in L(c';\Pi(N'))$:

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

Further, $\bar{\mu} \le 2^{Q_5(\#G)}$

for some fixed polynomial Q_5.

Proof of claim. From lemma 3.11 it follows that there are an element $c" \in C(N")$ and $\Pi = \{\pi_1,\dots,\pi_r\} \subset \Pi(N")$, $r \le \mathrm{card}(N")$,such that

$$c' + \sum_{j=1}^{r} \pi_j = c" + \sum_{l=1}^{t} \lambda_l \bar{\pi}_l ,$$

where $\lambda_1,\dots,\lambda_t \in \mathbb{N}_o$, $\bar{\pi}_1,\dots,\bar{\pi}_t \in \Pi(N")$.

Hence

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

and therefore $\gcd(\Pi(N'')) \mid |c'-c''|$.

Let $\bar{c}'$ and $\bar{c}''$ be the constants of $L(c';\Pi(N'))$, $L(c'';\Pi(N''))$ in the representations as u.p.sets constructed in lemma 4.4. Choosing

$$\bar{\mu} = \bar{\mu}(G) := \underset{N',N''}{\mathrm{Max}} \{ \mathrm{Max}\{\bar{c}', \bar{c}''\}\}$$

we obtain the first statement, since

$$\gcd(\Pi(N'')) \mid \gcd(\Pi(N'))$$

and

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

Because $\bar{\mu} \leq 2^{Q_2(\# G)}$, the second statement of the claim follows. This completes the proof of the claim. □

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

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

and

$$D(N'') := \{ v \in L(C(N'');\Pi(N'')) \mid \bar{\mu} \leq v < \bar{\mu}+p_{N''} \text{ if } \bar{\mu} \in L(C(N'');\Pi(N'')), \text{ otherwise } \bar{\mu} < v \leq \bar{\mu}+p_{N''} \} .$$

Then we have

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

where $F_G := \{ v \in \psi(L(G)) \mid \bigcup_{N''\in \mathfrak{M}(G)} L(D(N'');p_{N''}) \}$. This fact follows from the above claim.

On the other hand we have

$$\forall\, \Theta = \{\Theta_{i_1},\dots,\Theta_{i_k}\} \in \underline{A}(G) :\ p_{N''} \mid \gcd(\Pi(\Theta_{i_j})),$$

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

$$p_{N''} \mid p_G \quad \text{for all } N'' \in \mathfrak{M}(G).$$

Defining

$$C_G := \{ v \in \bigcup_{N''\in \mathfrak{M}(G)} L(D(N'');p_{N''}) \mid v \leq \bar{\mu}+p_G \}$$

the first statement of proposition 4.5 follows.

The fact that

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

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

From proposition 4.5 the following proposition follows.

__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 follows immediately from proposition 4.5. □

§5. The Complexity of INEQ

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

An alternating Turing machine M with at most one alternation 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 $\in \Sigma_2^p$. □

__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 ∪ 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 it is to 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 is in the set defined by γ, iff $0^n \in L(G(\gamma))$.

Since + corresponds to concatenation and ∪ corresponds to union of c.f.1LA 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 have

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

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

Acknowledgements. The author wishes to thank B. Becker and H.U. Simon for some critical remarks concerning this paper.

References.

[1] Fürer,M.:"The Complexity of the Inequivalence Problem for Regular Expressions with Intersection", Proc. of the 7th Colloq. on Automata, Languages and Programming, LNCS 85, pp.234-245, 1980.

[2] Ginsburg,S.:"The Mathematical Theory of Context-Free Languages", New York: McGraw-Hill Book Co.,1966.

[3] Hunt III,H.B.,D.J.Rosenkrantz & T.G. Szymanski:"On the Equivalence, Containment, and Covering Problems for the Regular and Context-Free Languages", J.Comp.&Syst.Scienc.,Vol.12,pp.222-268,1976.

[4] Huynh,Th.D.:"The Complexity of Semilinear Sets",to appear in Elektronische Informationsverarbeitung und Kybernetik 1982.

[5] Huynh,Th.D.:"Commutative Grammars: The Complexity of Uniform Word Problems", 1981. (Submitted for Publication.)

[6] Huynh,Th.D.:"Remarks on the Complexity of an Invariant of Context-Free Grammars", Acta Informatica,Vol.17,pp.89-99,1982.

[7] Rangel,J.L.:"The Equivalence Problem for Regular Expressions over One Letter is Elementary",15th Annual Symp. on Switching and Automata Theory,pp.24-27, 1974.

[8] Stockmeyer,L.J.:"The Complexity of Decision Problems in Automata Theory and Logic",Report TR-133,M.I.T,Project MAC, Cambridge,Mass.,1974.

[9] Stockmeyer,L.J. & A.R. Meyer:"Word Problems Requiring Exponential Time: Preliminary Report",Proc.of the 5th Annual ACM Symp.on the Theory of Computing, pp.1-9,1973.