INCLUSION OF REGULAR AND LINEAR LANGUAGES IN GROUP LANGUAGES

Krasimir Yordzhev

Faculty of Mathematics and Natural Sciences
South-West University, Blagoevgrad, Bulgaria
E-mail: yordzhev@swu.bg

Abstract
Let \( \Sigma = X \cup X^{-1} = \{ x_1, x_2, ..., x_m, x_1^{-1}, x_2^{-1}, ..., x_m^{-1} \} \) and let \( G \) be a group with set of generators \( \Sigma \). Let \( \mathcal{L}(G) = \{ \omega \in \Sigma^* \mid \omega \equiv e \pmod{G} \} \subseteq \Sigma^* \) be the group language representing \( G \), where \( \Sigma^* \) is a free monoid over \( \Sigma \) and \( e \) is the identity in \( G \). The problem of determining whether a context-free language is subset of a group language is discussed. Polynomial algorithms are presented for testing whether a regular language, or a linear language is included in a group language. A few finite sets are built, such that each of them is included in the group language \( \mathcal{L}(G) \) if and only if the respective context-free language is included in \( \mathcal{L}(G) \).

Key words: group language; context-free language; regular language; linear language; finite automaton; linear grammar; transition diagram

2010 AMS subject classifications: 68Q45; 68Q70

1 Introduction
For more information on automata and language theory we refer the reader to [6]. For the mathematical foundations and algebraic approach of formal language theory we refer to [7, 8]. For the connections between formal language theory and group theory we recommend the source [3]. List of open problems related to the discussed in this paper topics is given in [4].

Let \( \Sigma = X \cup X^{-1} = \{ x_1, x_2, ..., x_m, x_1^{-1}, x_2^{-1}, ..., x_m^{-1} \} \) (1) be an finite alphabet and let \( \Sigma^* \) denote the free monoid over \( \Sigma \). Let \( G \) be a group with the set of generators \( \Sigma \), the set of defining relations \( \Theta \), unit element \( e \) and with decidable word problem. Then the set of words

\[ \mathcal{L}(G) = \{ \omega \in \Sigma^* \mid \omega \equiv e \pmod{G} \} \subseteq \Sigma^* \] (2)

will be called a group language, which specifies the group \( G \). The group \( G \) is specified by a context-free language, if the relevant group language \( \mathcal{L}(G) \) is context-free. The group \( G \) in this case is called a context-free group.

The notion of group language was introduced by A. V. Anisimov in [1]. In this article Anisimov proved that \( \mathcal{L}(G) \) is regular if and only if the group \( G \) is finite (See also [3] Theorem 5.17).
A somewhat different definition of the term group language is given in [5], namely a regular language whose syntactic monoid is a finite group. In the given above definition are allowed context-free languages which are not regular. In our work we will stick to the first definition given by A.V. Anisimov.

In [2] A.V. Anisimov has showed that the problem of determining the unambiguity of finite automata is a special case of the problem of determining whether a context-free language is a subset of a group language. Then the question of finding a polynomial algorithms verifying the inclusion of context-free languages in group languages naturally arises. This article discusses the most important types of context-free languages - the regular and the linear ones. Regular languages are presented with the help of finite automata, and linear languages with the help of linear grammars. In both cases a few finite sets are built, such that each of them is included in the group language \( L(G) \) if and only if the respective context-free language is included in \( L(G) \). As a result polynomial algorithms verifying the inclusion of a regular, or a linear language in a group language are presented.

Throughout this article \( G \) will be a finitely generated group with decidable word problem, and \( \Gamma = (N, \Sigma, \Pi, A_1) \) will be a context-free grammar that generates the context-free language \( L \), ie \( L = L(\Gamma) \), where \( N \) is the set of variables (nonterminals), \( \Sigma = \{x_1, x_2, ..., x_m, x_{-1}^1, x_{-1}^2, ..., x_{-1}^m\} \) is the set of terminals, \( \Pi \) the set of productions, and \( A_1 \in N \) the start symbol. Let \( r \) be the constant of the pumping lemma for context-free language \( L \) (see [6, Theorem 7.18]).

We define the sets:
\[
\Omega_1 = \{\omega \in L \mid |\omega| \leq r\};
\]
\[
\Omega_2 = \{uwv^{-1} \mid |uwv| \leq r, uv \neq \varepsilon, \exists A \in N : A \xrightarrow{*} uAv, A \xrightarrow{*} w\};
\]
\[
W_1 = \Omega_1 \cup \Omega_2.
\]

The following theorem is proved in [2]:

**Theorem 1.1 (A. V. Anisimov [2])** With the above notation, \( L \subseteq \mathcal{L}(G) \) if and only if \( W_1 = \Omega_1 \cup \Omega_2 \subseteq \mathcal{L}(G) \).

Theorem 1.1 gives us an algorithm to check whether the inclusion \( L \subseteq \mathcal{L}(G) \) is true. Unfortunately, this algorithm is not polynomial. The purpose of this work is to show that if \( L \) is a regular or a linear language, then Anisimov’s algorithm can be transformed so as to be polynomial.

A **transition diagram** is a 4-tuple \( H = (V, R, S, l) \), where \( (V, R) \) is a directed graph with set of vertices \( V \) and multiset of arcs \( R \subseteq V \times V = \{(v_1, v_2) \mid v_1, v_2 \in V\} \); \( S \) is a semigroup whose elements will be called labels and \( l \) is a mapping from \( R \) to \( S \), which we call labeling mapping. If \( \pi = p_1 \cdot p_2 \cdot \cdots \cdot p_k \) is a walk in \( H \), \( p_i \in R \), \( i = 1, 2, \ldots, k \) then by definition
\[
l(p_1 \cdot p_2 \cdot \cdots \cdot p_k) = l(p_1)l(p_2)\ldots l(p_k).
\]

If \( P \) is a set of walks in \( H \), then by definition
\[
l(P) = \bigcup_{\pi \in P} l(\pi) = \{\omega \in S \mid \exists \pi \in P : l(\pi) = \omega\}
\]

Some of the outcomes in this article were announced in Russian in the conference [9].
2 Inclusion of regular languages in group languages

Throughout this section $L$ will mean a regular language. Then there is a (deterministic or nondeterministic) finite automaton $A = (Q, \Sigma, \delta, q_1, Z)$ such that

$$L = L(A) = \{ \omega \in \Sigma^* | \delta(q_1, \omega) \cap Z \neq \emptyset \},$$

where:
- $Q = \{ q_1, q_2, \ldots, q_n \}$ is the set of states;
- $\Sigma = \{ x_1, x_2, \ldots, x_m, x_1^{-1}, x_2^{-1}, \ldots, x_m^{-1} \}$ is the set of input symbols;
- $\delta$ is the transition function;
- $q_1 \in Q$ is the start state;
- $Z \subseteq Q$ is the set of final (or accepting) states.

Let $H_A = (Q, R, \Sigma^*, l_A)$ be the transition diagram for the automaton (3) (see [6, p. 48]). Let $G$ be a group with decidable word problem, with the set of generators $\Sigma = \{ x_1, x_2, \ldots, x_m, x_1^{-1}, x_2^{-1}, \ldots, x_m^{-1} \}$ and unit element $e$. Let $H_G = (Q, R, G, l_G)$ be the transition diagram with the same set of vertices and arcs as in $H_A$, but we consider the labels of arcs as elements of the group $G$.

We consider the semiring

$$F_G = (\mathcal{P}(G), \cup, \cdot, \phi, \{ e \}),$$

where $\mathcal{P}(G)$ is the set of subsets of $G$. Operations in $F_G$ are respectively the union and the product of sets, identity is the set $\{ e \}$ that contains only the identity $e$ of $G$, and zero - the empty set $\phi$.

Let $X, Y \in F_G$. In the semiring $F_G$ we define the next binary operation:

$$X \star Y = \{ xyx^{-1} | x \in X, y \in Y \}$$

We consider the following sets of walks in $H_G$:

- $P_{ij}$ - the set of all walks $\pi \in H_G$ with the initial vertex $q_i \in Q$ and the final vertex $q_j \in Q$, $1 \leq i, j \leq n$;
- $\widehat{P}_{ij}$ - the set of all walks $\pi \in H_G$ with the initial vertex $q_i \in Q$, the final vertex $q_j \in Q$, $1 \leq i, j \leq n$, and in which all vertices are distinct, except possibly $q_i = q_j$. $\widehat{P}_{ij} \subseteq P_{ij}$;
- $P_{iZ}$ - the set of all walks $\pi \in H_G$ with the initial vertex $q_i \in Q$ and the final vertex an element of $Z$, $1 \leq i \leq n$;
- $\widehat{P}_{iZ}$ - the set of all walks $\pi \in H_G$ with the initial vertex $q_i \in Q$, the final vertex an element of $Z$, $1 \leq i \leq n$, and in which all vertices are distinct (except possibly the initial and final vertices). $\widehat{P}_{iZ} \subseteq P_{iZ}$;
Oi – the set of all walks π ∈ H_G with the initial vertex and the final vertex q_i ∈ Q, 1 ≤ i ≤ n, and in which all vertices are distinct (except initial and final vertices which are q_i). O_i = P_{iZ}.

Obviously L ⊆ Λ(G) if and only if l_G(P_{iZ}) = {e}.

Let the equivalence of conditions (i) and (ii) was proved by A.V. Anisimov in [2] (iii) implies (i) and (i) implies (iv).

We consider the following elements of the semiring K:

\begin{align*}
\Omega_3 &= \{ l_G(\pi) \mid \pi \in P_{iZ} \} \\
\Omega_4 &= \{ w\pi l_G(\pi) \mid \exists q_j \in Q, \pi_1 \in P_{j1}, \pi_2 \in P_{j2} : u = l_G(\pi_1), v = l_G(\pi_2) \} \\
W_2 &= \Omega_3 \cup \Omega_4 \in F_G.
\end{align*}

We define the sets of walks \( K_{ij} \) in \( H_G \), where i, j ∈ \{1, 2, ..., n\}, k ∈ \{0, 1, ..., n\}, n = |Q| as follows:

\begin{align*}
K_{ij}^0 &= \{ \rho \mid \rho = (q_i, q_j) \in R \} \\
K_{ij}^k &= K_{ij}^{k-1} \cup K_{ik}^{k-1} K_{kj}^{k-1}.
\end{align*}

By definition \( K_{ij}^k \) consists only of walks with the initial vertex \( q_i \in Q \) the final vertex \( q_j \in Q \), and may not pass through a vertex \( q_k \) where \( s \geq k \) or that passes along a walk \( \pi_1 \) from \( q_i \) to \( q_k \), then passes along a walk \( \pi_2 \) from \( q_k \) to \( q_j \). None of these walks \( \pi_1 \) or \( \pi_2 \) passes along an interior vertex \( q_s \) where \( s > k \).

We consider the following elements of the semiring F_G:

\begin{align*}
\Omega_5 &= \{ l_G(\pi) \mid \pi \in K_{1s}, q_i \in Z \} \\
\Omega_6 &= \{ l_G(\pi_1) \ast l_G(\pi_2) \mid q_j \in Q, q_i \in Z : K_{ij}^n \neq \emptyset, \pi_1 \in K_{ij}^n, \pi_2 \in K_{jj}^n \}, \text{ where "\ast" is defined by \( \boxplus \) operation;}
W_3 &= \Omega_5 \cup \Omega_6 \in F_G.
\end{align*}

It is not difficult to see that

\begin{equation}
\Omega_3 \subseteq \Omega_5 \quad \text{and} \quad \Omega_4 \subseteq \Omega_6.
\end{equation}

As in \( K_{ij}^k \) is possible existence of a walk containing a cycle or a loop, then in the general case \( \Omega_3 \neq \Omega_5 \) and \( \Omega_4 \neq \Omega_6 \).

**Theorem 2.1** Let \( L \) be a regular language and let \( L = L(A) \), where \( A \) is defined by \( \boxplus \) automaton. Then with the above notation, the following conditions are equivalent:

(i) \( L \subseteq \Lambda(G) \)

(ii) \( W_1 = \Omega_1 \cup \Omega_2 = \{e\} \)

(iii) \( W_2 = \Omega_3 \cup \Omega_4 = \{e\} \)

(iv) \( W_3 = \Omega_5 \cup \Omega_6 = \{e\} \)

Proof. Since regular languages are special cases of context-free languages, the equivalence of conditions (i) and (ii) was proved by A.V. Anisimov in [2] (Theorem 1.1). Besides \( W_2 \subseteq W_3 \) (see (7)), i.e. \( W_3 = \{e\} \) implies \( W_2 = \{e\} \). So we proved that (iv) implies (iii). To prove the theorem we have to prove that (iii) implies (i) and (ii) implies (iv).

(iii) implies (i): Let \( W_2 = \Omega_3 \cup \Omega_4 = \{e\} \) and let \( \omega \in L \). Then there is a walk \( \pi \in P_{iZ} \) such that \( l_A(\pi) = \omega \).

If \( \pi \) does not contain cycles and loops, then \( l_G(\pi) \in l_G(P_{iZ}) = \Omega_3 = \{e\} \) and therefore \( \omega \in \Lambda(G) \).
Let π contains a cycle or a loop. In other words, there is $q_i \in Q$ such that
π can be expressed as $\pi = \pi_1 \pi_2 \pi_3$, where $\pi_1 \in P_{ij}, \pi_2 \in O_i, \pi_3 \in P_j$ and
$l_G(\pi_1)l_G(\pi_2)(l_G(\pi_1))^{-1} \in \Omega = \{e\}$. Therefore, $l_G(\pi_1)l_G(\pi_2) = l_G(\pi_1)$ and
$l_G(\pi_1 \pi_2 \pi_3) = l_G(\pi_1 \pi_3)$. Since $\pi_2 \in O_i$, then the length of $\pi_2$ is greater than 1.
Consequently, in $H_G$ there is a walk with less length than the length of $\pi$, whose label is equal to $\omega$ in the group $G$. This process of reduction may proceed a finite number of times as the length of $\omega$ is finite. At the end of this process we obtain a walk in $H_G$ without cycles and without loops with label equal to $\omega$ as an element of the group $G$. But $l_G(P_{ij}) = \Omega_3 = \{e\}$. Hence $\omega = e$ in the group $G$ and therefore $L \subseteq \Sigma(G)$.

(i) implies (iv): Let $L \subseteq \Sigma(G)$ be $l_G(P_{ij}) = \{e\}$. From $\Omega_5 \subseteq l_G(P_{ij})$ follows
$\Omega_5 = \{e\}$. Let $z \in \Omega_6$. Then $z$ can be represented in the form $z = uvu^{-1}$, where $u \in l_G(K_{j_i}^1), v \in l_G(K_{j_i}^n)$ for some integer $j$ such that there is a walk $\pi_3 \in P_{ij}$ and let $l_G(\pi_3) = w$. Obviously there are a walk $\pi_1 \in P_{ij}$ and a walk $\pi_2 \in O_i$ such that $u = l_p(\pi_1)$ and $v = l_p(\pi_2)$. Thus $\pi' = \pi_1 \pi_2 \pi_3 \in P_{ij}$ and $\pi'' = \pi_1 \pi_3 \in P_{ij}$. Since $L \subseteq \Sigma(G)$ then $l_G(\pi') = l_G(\pi'') = e$, therefore $uvw = uv, \text{ie} uvu^{-1} = e$. Hence $z = e$ and since $z \in \Omega_6$ is arbitrary, then $\Omega_6 = \{e\}$. The theorem is proved.

The following algorithm is based on the equivalence (i) and (iv) of Theorem 2.1

For convenience, $i \in Z$ will mean $q_i \in Z$, and $g_{ij}^k$ will be $l_G(K_{ji}^k)$. Here, $k$ in $g_{ij}^k$ is a superscript and does not mean an exponent.

**Algorithm 2.1** Verifies the inclusion $L \subseteq \Sigma(G)$ for a regular language $L$, and a group language $\Sigma(G)$, where $G$ is a group with decidable word problem.

**Input:** $g_{ij}^k = l_G(K_{ji}^k), \; i, j = 1, 2, \ldots, n$

**Output:** Boolean variable $T$, which receives the value True if $L \subseteq \Sigma(G)$, and the value False, otherwise. The algorithm will stop immediately after the value of $T := \text{False}$.

Begin
1. $T := \text{True}$;
2. For $1 \leq k \leq n$ Do
3. For $1 \leq i, j \leq n$ Do
4. $g_{ij}^k := g_{ij}^{k-1} \cup g_{ij}^{k-1} g_{ij}^{k-1}$;
5. End Do;
6. End Do;
7. For $j \in Z$ Do
8. If $g_{ij}^n \neq \phi$ and $g_{ij}^n \neq \{e\}$ Then
9. Begin $T := \text{False}$; Halt; End;
10. End Do;
11. For $1 \leq j \leq n$ Do
12. For $t \in Z$ Do
13. If $g_{ij}^n \neq \phi$ and $g_{ij}^n \neq \phi$ and $g_{ij}^n \neq \phi$ Then
14. If $g_{ij}^n \neq \{e\}$ Then
15. Begin $T := \text{False}$; Halt; End;
16. End Do;
17. End Do;
18. End Do;
End.
Theorem 2.2 Algorithm 2.1 checks the inclusion \( L \subseteq \mathcal{L}(G) \), where \( L \) is a regular language recognized by a finite automaton with \( n \) states, \( \mathcal{L}(G) \) is a group language, which specifies the group \( G \) with decidable word problem. Algorithm 2.1 executes at most \( O(n^3) \) operations \( \cup \) and \( \cdot \), and at most \( O(n^2) \) operations \( \star \) in the semiring \( F_G \), where the binary operation \( \star \) is defined using the formula (4).

Proof. According to Theorem 2.1 and considering axioms of the semiring \( F_G \), then in rows 9 and 15 of Algorithm 2.1, the boolean variable \( T \) gets the value \( \text{False} \) if and only if \( L \) is not included in \( \mathcal{L}(G) \). Otherwise, \( T \) gets the value \( \text{True} \). Hence the algorithm correctly checks whether the inclusion \( L \subseteq \mathcal{L}(G) \) is true.

It is easy to see that line 4 is executed no more than \( n^3 \) times. The operations \( \cup \) and \( \cdot \) (once each of them) in the semiring \( F_G \) is performed during each iteration. Lines 13 and 14 is executed at most \( n^2 \) times each. Therefore, Algorithm 2.1 performs no more than \( O(n^3) \) operations \( \cup \) and \( \cdot \), and no more than \( O(n^2) \) operations \( \star \) in the semiring \( F_G \). The theorem is proved. \( \square \)

Corollary 2.1 If the operations \( \cup, \cdot \) and \( \star \) in the semiring \( F_G \) can be done in a polynomial time, then Algorithm 2.1 is polynomial.

3 Inclusion of linear languages in group languages

Let \( S \) be an arbitrary monoid with identity 1. We consider the set 

\[ U_S = S \times S = \{(x, y) | x, y \in S\}. \]

We introduce the operation \( \diamond \) in \( U_S \) as follows: if \( (x, y), (z, t) \in U_S \) then

\[ (x, y) \diamond (z, t) = (xz, ty). \] (8)

It is easy to see that the operation \( \diamond \) is associative and \( U_S \) with this operation is a monoid with identity \((1,1)\). If \( S \) is a group, then \( U_S \) is a group, and if \( a = (x, y) \in U_S \) then the inverse element of \( a \) will be \( a^{-1} = (x^{-1}, y^{-1}) \). We define mappings \( f_l, f_r \) and \( f_d \) from \( U_S \) to \( S \) as follows:

\[ f_l(x, y) = x \] (9)

\[ f_r(x, y) = y \] (10)

\[ f_d(x, y) = xy \] (11)

Obviously

\[ f_d(x, y) = f_l(x, y)f_r(x, y). \]

In this section we consider a linear grammar

\[ \Gamma = (N, \Sigma, \Pi, A_1), \] (12)

where:

\( N = \{A_1, A_2, \ldots, A_n\} \) is the set of variables (nonterminals);

\( \Sigma = \{x_1, x_2, \ldots, x_m, x_1^{-1}, x_2^{-1}, \ldots, x_m^{-1}\} \) is the set of input symbols;

\( \Pi \) is the set of productions;
A$_1 \in N$ is the start variable.

A context-free grammar $\Gamma = (N, \Sigma, \Pi, A_1)$ is called linear if all productions in $\Pi$ are of the form $A_i \rightarrow \alpha A_j \beta$ or $A_i \rightarrow \alpha$, where $A_i, A_j \in N$, $1 \leq i, j \leq n$, $\alpha, \beta \in \Sigma^*$. A language $L$ is called linear if there is a linear grammar $\Gamma$ such that $L = L(\Gamma, A_1)$.

We consider the transition diagram

$$H_\Gamma = (V, R, U_\Sigma, l_\Gamma)$$

with the set of vertices $V = N \cup \{A_{n+1}\}$, where $A_{n+1} \notin N$. $U_\Sigma^*$ is the considered above monoid with the set of elements $\{(\alpha, \beta)|\alpha, \beta \in \Sigma^*\}$ and with the operation $\circ$. The set of arcs $R$ in $H_\Gamma$ is formed as follows:

a) if a production $A_i \rightarrow \alpha A_j \beta$ exists in $\Pi$ where $A_i, A_j \in N$, then there exists an arc from $A_i$ to $A_j$ labeled $(\alpha, \beta)$;

b) if a production $A_i \rightarrow \alpha$ exists in $\Pi$ where $A_i \in N$, $\alpha \in \Sigma^*$, then there exists an arc from $A_i$ to $A_{n+1}$ labeled $(\alpha, \varepsilon)$, $\varepsilon$ is the empty word;

c) there are no other arcs in $R$.

Let $G$ be a group with the set of generators $\Sigma = \{x_1, x_2, ..., x_m, x_1^{-1}, ..., x_m^{-1}\}$, with the set of defining relations $\Theta$, identity $e$ and with decidable word problem. Let $U_G$ be the group obtained as described above. We consider the transition diagram

$$H_U = (V, R, U_\Sigma, l_U),$$

where the set of vertices $V$ and the set of arcs $R$ coincide with the corresponding sets in the transition diagram $H_\Gamma$ according to (13), and labels will be elements of the group $U_G$.

We consider the following sets of walks in $H_U$:

$D_{ij}$ – the set of all walks $\pi \in H_U$ with the initial vertex $A_i \in V$ and the final vertex $A_j \in V$, $1 \leq i \leq n$, $1 \leq j \leq n + 1$;

$\overline{D}_{ij}$ – the set of all walks $\pi \in H_U$ with the initial vertex $A_i \in V$, the final vertex $A_j \in V$, $1 \leq i \leq n$, $1 \leq j \leq n + 1$, and in which all vertices are distinct, except possibly $A_1 = A_j$. $\overline{D}_{ij} \subseteq D_{ij}$;

$C_i$ – the set of all walks $\pi \in H_U$ with the initial vertex and the final vertex $A_i \in V$, $1 \leq i \leq n$, and in which all vertices are distinct (except initial and final vertices which are $A_i$). $C_i = \overline{D}_n$.

**Lemma 3.1** Let $\Gamma = (N, \Sigma, \Pi, A_1)$ be a linear grammar and $H_\Gamma$ be the transition diagram according to (13). Let $P_\Gamma$ be the set of all walks $\pi \in H_\Gamma$ with the initial vertex $A_1$ and the final vertex $A_{n+1}$ Then

$$L = L(\Gamma) = f_d(l_\Gamma(P_\Gamma)).$$

**Proof.** Immediate. \hfill \Box

**Corollary 3.1** Let $L$ be a linear language generated by the linear grammar (13) and let $G$ be a finitely generated group with decidable word problem and with the set of generators $\Sigma = \{x_1, x_2, ..., x_m, x_1^{-1}, ..., x_m^{-1}\}$. Then

$$L \subseteq \mathcal{L}(G) \iff f_d(l_U(D_{1,n+1})) = \{e\}.$$
As in section 2, we can consider the semirings \( F_G = (\mathcal{P}(G), \cup, \cdot, \phi, \{e\}) \) and \( F_U = (\mathcal{P}(U_G), \cup, \circ, \phi, \{(e, e)\}) \). Defined by using equations (9), (10) and (11) mappings \( f_L, f_r, f_d \) can be extended in a natural way to mappings from \( F_U \) to \( F_G \).

Let \( X, Y, Z \in F_G \). In \( F_G \), we introduce the next operation:

\[
\langle X, Y, Z \rangle = \{xzy^{-1} \mid x \in X, y \in Y, z \in Z\} \tag{15}
\]

We consider the following elements of the semiring \( F_G \):

\[
\Omega_7 = \left\{ f_d(l_U(\pi)) \mid \pi \in D_{1n+1} \right\};
\]

\[
\Omega_8 = \left\{ (f_l(l_U(\pi_2)), f_d(l_U(\pi_3)), f_r(l_U(\pi_2))) \mid \exists \pi \in D_{1n+1} : \pi = \pi_1\pi_2\pi_3, \right. \\
\left. \pi_1 \in D_{1i}, \pi_2 \in C_i, \pi_3 \in D_{i+1}, 1 \leq i \leq n \right\};
\]

\[
W_4 = \Omega_7 \cup \Omega_8.
\]

It is not difficult to see that

\[
\Omega_7 \subseteq \Omega_1 \quad \text{and} \quad \Omega_8 \subseteq \Omega_2 \implies W_4 \subseteq W_1 \tag{16}
\]

and in the general case \( \Omega_7 \neq \Omega_1 \) and \( \Omega_8 \neq \Omega_2 \).

As in section 2 (see (12) and (13)), we define the sets of walks \( K_{ij}^k \) in \( H_U \), where \( i \in \{1, 2, \ldots, n\}, j \in \{1, 2, \ldots, n+1\}, k \in \{0, 1, \ldots, n\}, n = |V|, V = N \cup \{A_{n+1}\}, N = \{A_1, A_2, \ldots, A_n\} \) is the set of variables of the grammar \( \Gamma \), \( A_{n+1} \notin N \).

\[
K_{ij}^0 = \{\rho \mid \rho = (A_i, A_j) \in R\} \tag{17}
\]

\[
K_{ij}^k = K_{ij}^{k-1} \cup K_{ik}^{k-1} K_{kj}^{k-1} \tag{18}
\]

Let \( g_{ij}^k = l_U(K_{ij}^k) \in F_U \), where \( k \) is a superscript and does not mean an exponent.

We consider the following elements of the semiring \( F_G \):

\[
\Omega_9 = \{ f_d(g_{ij}^n) \};
\]

\[
\Omega_{10} = \left\{ (f_l(g_{ij}^n), f_d(g_{ij}^r), f_r(g_{ij}^l)) \mid 1 \leq i \leq n, K_{ij}^n \neq \emptyset, K_{ij}^n \neq \emptyset, K_{ij}^{n+1} \neq \emptyset \right\};
\]

\[
W_5 = \Omega_9 \cup \Omega_{10}.
\]

It is easy to see that

\[
\Omega_7 \subseteq \Omega_9 \quad \text{and} \quad \Omega_8 \subseteq \Omega_{10} \implies W_4 \subseteq W_5. \tag{19}
\]

As in \( K_{ij}^k \) is possible existence of a walk containing a cycle or a loop, then in the general \( \Omega_7 \neq \Omega_9 \) and \( \Omega_8 \neq \Omega_{10} \).

**Theorem 3.1** Let \( L \) be a linear language and let \( L = L(\Gamma) \), where \( \Gamma \) is defined by (12) linear grammar. Then with the above notation, the following conditions are equivalent:

(i) \( L \subseteq \Omega(G) \);

(ii) \( W_1 = \Omega_1 \cup \Omega_2 = \{e\} \);

(iii) \( W_4 = \Omega_7 \cup \Omega_8 = \{e\} \);

(iv) \( W_5 = \Omega_9 \cup \Omega_{10} = \{e\} \).

Proof. The equivalence of conditions (i) and (ii) was proved by A.V. Anisimov in (2) (Theorem 11). As noted above (see (14)), \( W_4 \subseteq W_1 \) and hence \( W_1 = \{e\} \) implies \( W_4 = \{e\} \), if (ii) implies (iii). From (19) follows that (iv) implies (iii). To prove the theorem it is sufficient to prove that (iii) implies (i) and (i) implies (iv).
(iii) implies (i): Let \( W_4 = \Omega_7 \cup \Omega_8 = \{e\} \) and let \( \omega \in L \). According to Lemma 3.1, \( \omega \in f_d(l_U(P_1)) \). Hence \( \omega \) can be written as \( \omega = \omega_1 \omega_2 \), where \( (\omega_1, \omega_2) \) is the label of a walk in \( H_T \) with the initial vertex \( A_1 \) and the final vertex \( A_{n+1} \) and let \( \pi \) be the corresponding path in \( H_U \). \( f_d(l_U(\pi)) \equiv \omega \) (mod \( G \)) is satisfied.

If \( \pi \in \overrightarrow{D_1}_{n+1} \) then \( f_d(l_U(\pi)) \in f_d(l_U(D_1_{n+1})) = \Omega_T \subseteq \{e\} \) and therefore \( \omega \in \Sigma(G) \).

Suppose \( \pi \) contains a cycle or a loop. Then \( \pi \) can be written as \( \pi = \pi_1 \pi_2 \pi_3 \), where \( \pi_1 \in D_{ij} \), \( \pi_2 \in C_i \) and \( \pi_3 \in \overrightarrow{D}_{n+1} \) for some \( j \in \{1, 2, \ldots, n\} \). Let \( l_U(\pi_1) = (a_1, b_1), l_U(\pi_2) = (a_2, b_2) \) and \( l_U(\pi_3) = (a_3, b_3) \). Then

\[
f_d(l_U(\pi)) = f_d((a_1, b_1) \circ (a_2, b_2) \circ (a_3, b_3)) = f_d(a_1a_2a_3, b_3b_2b_1) = a_1a_2a_3b_3b_2b_1.
\]

But \( a_2a_3b_2(a_3b_3)^{-1} \in \Omega_8 \), hence \( a_2a_3b_2(a_3b_3)^{-1} = e \), i.e. \( a_1a_2a_3b_3b_2b_1 = a_1a_3b_3b_1 \). It is easy to see that \((a_1a_3, b_3b_1)\) is the label of the walk \( \pi_1\pi_3 \), which is obtained from \( \pi \) by omitting \( \pi_2 \). We continue to omit the cycles and loops in \( \pi \). Because the word \( \omega \) is finite, after finitely many steps we obtain a walk \( \pi' \in \overrightarrow{D}_{n+1} \) such that \( f_d(l_u(\pi')) = f_d(l_U(\pi)) \equiv \omega \) (mod \( G \)). But \( f_d(l_U(\pi')) \in \Omega_T \subseteq \{e\} \) and according to Corollary 3.1, \( L \subseteq \Omega(G) \).

(i) implies (iv): Let \( L \subseteq \mathcal{M} \). Then according to Corollary 3.1, \( f_d(l_U(D_1_{n+1})) = \{e\} \). It is obvious that \( K^n_{1, n+1} \subseteq D_{1, n+1} \) and therefore \( \Omega_9 = \{e\} \).

Let \( z \in \Omega_{10} \). Then \( z = uvv^{-1} \), where \( u \in f_1(g^n_{1i}), v = f_d(g^n_{1i+1}), w = f_d(g^n_{1i}) \) and there are walks \( \pi_1 \in D_{1i}, \pi_2 \in C_i, \pi_3 \in D_{n+1} \) for some \( i \in \{1, 2, \ldots, n\} \) such that \((u, w) = l_u(\pi_2) \) and \( v = v_1v_2, \) where \((v_1, v_2) = l_u(\pi_3) \).

Let \( l_U(\pi_1) = (x, y) \). We consider the walks \( \pi' = \pi_1\pi_2\pi_3 \in D_{1, n+1} \) and \( \pi'' = \pi_1\pi_3 \in D_{1, n+1} \). We have:

\[
\begin{align*}
l_U(\pi') &= l_U(\pi_1\pi_2\pi_3) = (x, y) \circ (u, w) \circ (v_1, v_2) = (xvv_1, v_2xy) \\
l_U(\pi'') &= l_U(\pi_1\pi_3) = (x, y) \circ (v_1, v_2) = (xv_1, v_2y)
\end{align*}
\]

According to Corollary 3, \( xvv_1 = xvy = e \), which implies that \( uvv^{-1} = e \), i.e. \( z = e \). Since \( z \in \Omega_{10} \) is an arbitrary \( z \), then \( \Omega_{10} = \{e\} \). The theorem is proved.

The following algorithm is based on the equivalence (i) and (iv) of Theorem 3.1

**Algorithm 3.1** Verifies the inclusion \( L \subseteq \Omega(G) \) for a linear language \( L \), and a group language \( \Omega(G) \), where \( G \) is a group with decidable word problem.

**Input:** \( g^n_{1i} = l_U(K^n_{ij}), i = 1, 2, \ldots, n, j = 1, 2, \ldots, n, n + 1 \)

**Output:** Boolean variable \( T \), which receives the value \( \textbf{True} \) if \( L \subseteq \Omega(G) \), and the value \( \textbf{False} \), otherwise. The algorithm will stop immediately after the value of \( T := \textbf{False} \).

```
begin
1. \( T := \textbf{True} \);
2. For 1 \( \leq k \leq n \) Do
3.   For 1 \( \leq i \leq n \) and 1 \( \leq j \leq n + 1 \) Do
4.     \( g^k_{ij} := g^{k-1}_{ij} \cup g^{k-1}_{kj} \circ g^{k-1}_{kj}; \)
5.   End Do;
6. End Do;
7. If \( g^n_{1n+1} \neq \emptyset \) \( \& \) \( f_d(g^n_{1n+1}) \neq \{e\} \) Then
8.   Begin \( T := \textbf{False}; \) Halt; End;
```
9. For $1 \leq i \leq n$ Do
10. If $g^n_i \neq \phi$ and $g^n_{i+1} \neq \phi$ and $g^n_{i+1} \neq \phi$ Then
11. If $\langle f_l(g^n_i), f_d(g^n_{i+1}), f_r(g^n_i) \rangle \neq \{e\}$ Then
12. Begin $T := \text{False}$; Halt; End;
13. End Do;
End.

**Theorem 3.2** Algorithm 3.1 checks the inclusion $L \subseteq \Sigma(G)$, where $L$ is a linear language generated by a linear grammar with $n$ variables, $\Sigma(G)$ is a group language, which specifies the group $G$ with decidable word problem. Algorithm 3.1 executes at most $O(n^3)$ operations $\cup$ and $\diamond$ in the semiring $F_U$, and no more than $O(n^2)$ operations $\langle , , \rangle$ in the semiring $F_G$, where the operation $\langle , , \rangle$ is defined using the formula (15).

Proof. Similarly as the proof of Theorem 2.2. \hfill \square

**Corollary 3.2** If the operations $\cup$, $\diamond$ in the semiring $F_U$ and the operation $\langle , , \rangle$ in the semiring $F_G$ can be done in a polynomial time, then Algorithm 3.1 is polynomial.

**References**

[1] A. V. Anisimov, Group languages, Cybernetics and Systems Analysis, 7 (1971), pp. 594–601.

[2] A. V. Anisimov, Finite-automaton semigroup mappings, Cybernetics and Systems Analysis, 17 (1981), pp. 571–578.

[3] I. Chiswell, A course in formal languages, automata and groups, Springer-Verlag, 2009.

[4] R. I. Grigorchuk, V. V. Nekrashevich, and V. I. Sushchanskii, Automata, dynamical systems, and groups, Proceedings of the Steklov Institute of Mathematics, 231 (2000), pp. 128–203. (Translated from Trudy Matematicheskogo Instituta imeni V.A. Steklova, Vol. 231, 2000, pp. 134-214).

[5] P.-C. Héam, On the complexity of computing the profinite closure of a rational language, Theoretical Computer Science, 412 (2011), pp. 5808 – 5813.

[6] J. E. Hopcroft, R. Motwani, and J. D. Ullman, Introduction to automata theory, languages, and computation, Addison-Wesley, 2 ed., 2001.

[7] G. Lallement, Semigroups and combinatorial applications, John Wiley & Sons, New York-Chichester-Brisbane, 1979.

[8] D. Perrin and J.-E. Pin, Infinite words. Automata, semigroups, logic and games, Elsevier, 2004.

[9] K. Yordzhev, Still on the problem of inclusion of regular and linear languages in group languages, in Algebra, Logic & Discrete mathematics, Niš University, April 14-16, 1995, pp. 699–710. in Russian.