GROUP LANGUAGES

A. V. Anisimov

UDC 51:621.391

Any group G with n generatrices $x_1, x_2, \ldots, x_n$ can be assigned as a homomorphic image of a free semigroup in the alphabet $\Sigma = \{x_1, x_2, \ldots, x_n, x_1^{-1}, x_2^{-1}, \ldots, x_n^{-1}\}$. Let φ be this homomorphism $\Sigma^* \to G$ and let $\mathfrak{M} = Ker\,\varphi$. By assigning the group G with the aid of defining relations, we completely determine the set $\mathfrak{M}$. It is well known that if G is finitely defined, then $\mathfrak{M}$ will be recursively enumerable and hence it will be a language of type 0 (see, for example, [1]). We shall say that the language $\mathfrak{M}$ specifies the group G.

Any language $\mathfrak{M}$ in an alphabet Σ that specifies a group in the above sense will be called a group language.

In this paper we shall study regular and context-free group languages. The properties of such languages are closely linked to the properties of the groups specified by them; for this reason their study reduces to an investigation of the corresponding groups.

It is proved that the class of groups specified by regular languages coincides with the class of finite groups. In the context-free case it was possible to fully describe only the class of abelian groups.

Let us note that group languages offer numerous examples of various types of languages on which it is convenient to verify the general assertions.

Suppose that a semigroup Π is specified by the generatrices $x_1, x_2, \ldots, x_n$ and the defining relations

$$A_i = B_i, \quad i = 1, 2, \ldots, m. \tag{1}$$

An elementary transformation of the semigroup Π is a transition from a word of the form XA_iY to a word XB_iY or conversely, where X and Y are words of the semigroup Π, and $A_i = B_i$ is one of the defining relations (1). Elementary transformations are expressed by the schemes

$$XA_iY \to XB_iY, \qquad XB_iY \to XA_iY.$$

The relations (1) define the equality of words in the semigroup Π. The words X and Y of a semigroup Π are equal in Π if and only if there exists a sequence of elementary transformations

$$X \to X_1 \to \ldots \to X_k \to Y$$

that carries the word X into the word Y.

A group G with generatrices $x_1, \ldots, x_n$ is defined as a semigroup specified by an alphabet $\Sigma = \{x_1, \ldots, x_n, x_1^{-1}, \ldots, x_n^{-1}\}$ and a system of defining relations

$$\begin{aligned} &A_i = 1, \qquad i = 1, 2, \ldots, m; \\ &x_j x_j^{-1} = 1, \\ &\quad x_j^{-1} x_j = 1, \quad j = 1, 2, \ldots, n. \end{aligned} \tag{2}$$

A free semigroup in the alphabet Σ will be denote by Σ^*. An equality relation defines a congruence ρ on $\Sigma^* \cdot X\rho Y$ if and only if $X = Y$ in Π. Hence there exists a natural homomorphism of Σ^* onto the group G. It is evident that this homomorphism can be realized through the free group generated by the set $X = \{x_1, x_2, \ldots, x_n\}$. Let $\mathfrak{M} = Ker\varphi \cdot \mathfrak{M}$ be a language in the alphabet Σ, since a language is by definition any subset

Translated from Kibernetika, No. 4, pp. 18-24, July-August, 1971. Original article submitted August 25, 1970.

of Σ^*. As was noted above, if G is finitely defined, then $\mathfrak{M}$ will be a language of type 0. On the basis of the defining relations (2) it is easy to construct a grammar that specifies the language $\mathfrak{M}$.

Let $\Gamma = (\Sigma, N, s, P)$ be a grammar, where Σ is a set of terminals, N a set of nonterminals, $s \in N$, s an initial symbol, and P are productions. By $L(\Gamma)$ we shall denote the language generated by this grammar. We shall say that a language $L(\Gamma)$ specifies a group G if in the above notations $L(\Gamma) = \mathfrak{M}$. In such a case the language $\mathfrak{M}$ is said to be a group language.

In the following we shall need the concept of a strong group automaton.

Let G be a group defined by the alphabet $\Sigma = \{x_1, \ldots, x_n, x_1^{-1}, \ldots, x_n^{-1}\}$. On the basis of the group G it is possible to construct a special automaton without outputs $A_G = (\Sigma, G, \delta)$. The states of A_G are elements of the group G. The transition function δ on the alphabet is defined as follows: $\delta(g, x_i) = g \circ \varphi(x_i)$, $\delta(g, x_i^{-1}) = g \circ \varphi(x_i^{-1})$, where φ is a natural homomorphism of Σ^* onto G, and $\circ$ is a group operation in G. The function δ can be continued in a natural way on words belonging to Σ^*. A strong group automaton is nothing else but an automaton interpretation of a colored Cayley graph. A strong group automaton is defined for arbitrary groups, not necessarily finite. Some properties of strong group automata can be found in [3].

Let S_g be a set of words in Σ^* that carry the automaton A_G from a state g into the same state g. It is easy to see that $S_g = \mathfrak{M}$ for $g \in G$.

Now let us study groups specified by regular languages.

THEOREM 1. Let $\Sigma = \{x_1, x_2, \ldots, x_n, x_1^{-1}, x_2^{-1}, \ldots, x_n^{-1}\}$ be an alphabet of the group G, let the language $\mathfrak{M}$ in the alphabet Σ specify a group G, i.e., there exists a homomorphism φ of Σ^* onto G, and let $\mathfrak{M} = Ker\,\varphi$. The group G is finite if and only if $\mathfrak{M}$ is a regular language.†

Proof. If G is a finite group, we shall consider a strong group automaton A_G. Since it is finite, the set $\mathfrak{M} = S_g$ will be regular.

Now let us assume that $\mathfrak{M}$ is a regular language. We shall prove a slightly more general result whence the assertion of the theorem can be obtained in an obvious manner.

If R is a regular event and the language $\mathfrak{M}$ specifies a group G and $R \subseteq \mathfrak{M}$, then R will carry any state g of the automaton A_G into itself by using only a finite number of states.

Let us begin with the following remark. If a regular notation of the event R contains $\{T\}$, where T is a regular event and $\{\,\}$ are iteration brackets, then $T \subseteq \mathfrak{M}$. In fact, let p be a word belonging to T. Since $\{T\}$ occurs in a regular notation of the event R, there exists a word $f_1 p f_2 \in R$, where f_1 and f_2 are words belonging to Σ^*. But in this case also the word $f_1 pp f_2 \in R$. Hence $\varphi(f_1 p f_2) = \varphi(f_1 pp f_2) = 1$, since $R \subseteq \mathfrak{M} = Ker\,\varphi$. Let us recall that φ denotes a natural epimorphism of Σ^* onto G. Hence $\varphi(f_1)\varphi(p)\varphi(f_2) = \varphi(f_1)\varphi^2(p)\varphi(f_2)$. After cancellations (which can be performed in view of the fact that φ maps Σ^* onto a group) we obtain $\varphi(p) = 1$, where 1 is the unit element of the group G, and hence $p \in \mathfrak{M}$. Since p is a word belonging to T, it follows that $T \subseteq \mathfrak{M}$.

The cyclic depth of a regular expression is defined as the maximum number of mutually embedded pairs of iteration brackets.

The cyclic depth of a regular event is the minimal cyclic depth occurring in a regular notation of an event.

The above assertion will be proved by induction on the cyclic depth.

Let us assume that R has a cyclic depth not greater than 1. By using the distributivity of multiplication for disjunctions, it is possible to write R in the form of a union of events T_i, $R = \vee T_i$, where T_i is

$$T_i = p_1 \{R_1\} p_2 \{R_2\} p_3 \ldots \{R_k\} p_{k+1},$$

p_j are words belonging to Σ^*, and R_j are events of depth 0. Since $R \subseteq \mathfrak{M}$, it follows that also $T_i \subseteq \mathfrak{M}$, and (as was shown above) $R_j \subseteq \mathfrak{M}$, $j = 1, 2, \ldots, k$, each R_i carrying any state g of a strong group automaton A_G into itself by using only a finite number of states, since R_j is a finite disjunction of words belonging to Σ^*.

†When this paper was already in print, A. A. Letichevskii was so kind to make the author aware of another proof of this theorem based on V. M. Glushkov's results concerning automatic partitions.

It is evident that $\{R_j\}$ likewise carries g into itself without adjoining any new states. The word p_1 carries g into g_1 after finitely many steps, i.e., in this transition the automaton A_G can be only in finitely many states, and $\{R_1\}$ carries g into itself by likewise using finitely many states. Under the action of p_2 the automaton goes over from the state g_1 into the state g_2, with the number of intermediate states remaining finite, etc. Hence follows that T_i carries any state g into itself by using only finitely many intermediate states. This reasoning is illustrated by the diagram

$$\begin{tikzcd} g \arrow[r, "p_1"] & g_1 \arrow[loop above, "\{R_1\}"] \arrow[r, "p_2"] & g_2 \arrow[loop above, "\{R_2\}"] \arrow[r, "p_3"] & \cdots \arrow[r, "p_k"] & g_k \arrow[loop above, "\{R_k\}"] \arrow[r, "p_{k+1}"] & g \end{tikzcd}$$

Some g_j may coincide, but this has no effect on the proof.

Since any T_i carries g into itself by using only finitely many states, it is evident that $R = VT_i$ will likewise carry g into itself by using finitely many states.

Suppose that the assertion is true for all regular events with a cyclic depth smaller than or equal to n, and let R have a cyclic depth $n+1$. By using (as above) the distributivity of multiplication with respect to disjunction, it is possible to express R in the form of a disjunction of finitely many events T_i, $R = VT_i$, where the T_i have the form

$$T_i = p_1 \{R_1\} p_2 \{R_2\} p_3 \ldots \{R_k\} p_{k+1},$$

and the R_j are regular events of cyclic depth not larger than $n-1$, $j = 1, 2, \ldots, k$. By reasoning in the same way as above and by using the induction hypothesis, we find that R carries any state g of the automaton A_G into itself by using only finitely many states, and thus we have proved the above assertion.

Hence we can see that if $\mathfrak{M}$ is regular, then $\mathfrak{M}$ will carry any state g into itself by using only finitely many intermediate states, and this signifies that the automaton A_G is finite, since in A_G it is possible to carry any state g into itself by traversing any state. Hence the group G is also finite, which completes the proof of the theorem.

This theorem can be regarded as a criterion of nonregularity of certain languages. Let us consider, for example, the grammar

$$\Gamma_{D_{2n}} = (\Sigma, s, s, P),$$

where $\Sigma = \{x_1, x_2, \ldots, x_n, x_1^{-1}, \ldots, x_n^{-1}\}$, and P denotes the following productions:

$$\begin{aligned} &s \to \varepsilon \\ &s \to ss \\ &s \to s x_i s x_i^{-1} s \\ &s \to s x_i^{-1} s x_i s, \quad i = 1, 2, \ldots, n. \end{aligned}$$

The symbol ε will always denote [illegible] ıpty word.

The language generated by this grammar is called Dick's language and denoted by D_{2n}. It is easy to show that Dick's language specifies a free group of rank n with free generatrices $x_1, x_2, \ldots, x_n$ [4]. In the theory of context-free languages it is well known that the language D_{2n} is not regular. From the point of view of group languages the nonregularity of D_{2n} follows directly from the infinity of a free group.

Another example: let $\Sigma = \{x_1, \ldots, x_n, x_1^{-1}, \ldots, x_n^{-1}\}$, $n > 1$. $\mathfrak{M}$ is a set of words p belonging to Σ^* such that the sum of the powers with which x_i occurs in p is equal to the sum of the powers of x_i^{-1} in this same word, $i = 1, 2, \ldots, n$. It is easy to show that $\mathfrak{M}$ specifies a factor group of a free group of rank n with respect to its commutator. This group is infinite, and hence the language $\mathfrak{M}$ is not regular. It follows from the results obtained below that $\mathfrak{M}$ is not even a context-free language.

Now let us study the groups specified by context-free (CF) languages.

As we noted above, in addition to finite groups this class contains also free groups of finite rank; however, Theorem 2 shows that this class is much wider.

THEOREM 2. The class of groups specified by CF languages is closed under the free-product operation.

Proof. Let G_1 and G_2 be groups specified by the CF grammars $\Gamma = (\Sigma_1, N_1, s_1, P_1)$ and $\Gamma_2 = (\Sigma_2, N_2, s_2, P_2)$. Let us construct a context-free grammar Γ_3 that specifies a free product of groups G_1 and G_2. It is always possible to assume that the alphabets of the groups G_1 and G_2 and of the grammars Γ_1 and Γ_2 are nonintersecting. Let $s \not\in \Sigma_1 \cup \Sigma_2 \cup N_1 \cup N_2$. If Y is a word in an alphabet, we shall define Y(s) as a word obtained from Y by insertions after each letter and before the first symbol s. For example, if $Y = x_1^{-1} \cdot x_2 N x_2$, then $Y(s) = s x_1^{-1} s x_2 s N s x_2 s$. Let us define $\Gamma(s)$ as a grammar obtained from Γ by adjoining the symbol s to nonterminals and replacing all the words Y in the right-hand sides by productions by Y(s). Let us consider the grammar $\Gamma_3 = (\Sigma_1 \cup \Sigma_2, N_1 \cup N_2 \cup s, P)$:

$$\Gamma_3 = \begin{cases} \Gamma_1 & (s) \\ \Gamma_2 & (s) \\ s \to \varepsilon \\ s \to s_1 \\ s \to s_2. \end{cases}$$

We shall assert that Γ_3 specifies $G_1 * G_2$.

Let $\mathfrak{M}_1 = L(\Gamma_1)$ be a language specifying G_1, let $\mathfrak{M}_2 = L(\Gamma_2)$ be a language specifying G_2, and $\mathfrak{M}$ a language in the alphabet $\Sigma_1 \cup \Sigma_2$ that specifies $G_1 * G_2$.

It is evident that $L(\Gamma_3) \subset \mathfrak{M}$. Now let $p \in \mathfrak{M}$. The proof that $p \in L(\Gamma_3)$ will be carried out by induction on the length of the word $p|p|$. For $|p| = 0$ we have $p = \varepsilon$, and the assertion is obvious. Let us assume that $|p| \neq 0$ and for all $q \in \mathfrak{M}$ we have $|q| < |p|$, $q \in L(\Gamma_3)$. If $p \in \mathfrak{M}$, then $p = U_1 V_1 U_2 V_2 \ldots U_t V_t$, where $U_i \in \Sigma_1^*$, $V_i \in \Sigma_2^*$, $i = 1, \ldots, t$ and the internal words are not empty. Since the value of the word p in $G_1 * G_2$ is equal to 1, it follows from the properties of a free product that there exists at least one U_i or V_i such that $U_i \in \mathfrak{M}_1$ or $V_i \in \mathfrak{M}_2$, and the corresponding U_i or V_i is not empty. We shall assume, for example, that $U_i \in \mathfrak{M}_1$. Let $p' = U_1 V_1 \ldots V_{i-1} V_i U_{i+1} V_{i+1} \ldots U_t V_t$. Let $p' \in \mathfrak{M}$ and $|p'| < |p|$. By using the induction hypothesis, we can assert that $s \Rightarrow p'$. It is easy to see that if $s \Rightarrow f_1 f_2$ in Γ_3, then also $s \Rightarrow f_1 s f_2$, where f_1 and f_2 are words. By using this property, we find that if $s \Rightarrow U_1 V_1 \ldots U_{i-1} V_{i-1} V_i U_{i+1} V_{i+1} \ldots U_t V_t$, then $s \Rightarrow U_1 V_1 \ldots V_{i-1} s v_i u_{i+1} V_{i+1} \ldots U_t V_t$, and since $s \to s_1$ and $s \Rightarrow U_i$, then also $s \Rightarrow U_1 V_1 \ldots V_{i-1} U_i V_i \ldots U_t V_t = p$. Hence $p \in L(\Gamma_3)$, which completes the proof of the theorem.

One and the same group can be assigned in different manners with the aid of generating elements and defining relations. It follows from Theorem 1 that irrespective of the manner in which we assign a finite group, the corresponding set $\mathfrak{M}$ will remain a regular event. A similar result holds for groups specified by CF languages.

THEOREM 3. The property of a finitely defined group to be specified by a CF grammar does not depend on the manner of assignment of this group by generating elements and defining relations.

Proof. If the group G is assigned in the alphabet $\Sigma = \{x_1, \ldots, x_n, x_1^{-1}, \ldots, x_n^{-1}\}$, $w(x) \in \Sigma^*$ and $b \not\in \Sigma$, it is possible to assign the group G by the alphabet $\Sigma \cup b \cup b^{-1}$ and by defining relations that are obtained from the previous relations by adjoining the defining relation $bw(x) = 1$, $bb^{-1} = 1$, $b^{-1}b = 1$.

On the other hand it is possible to eliminate b and b^{-1} from the system of generatrices if, in addition to the relations $bb^{-1} = 1$ and $b^{-1}b = 1$, b or b^{-1} occurs only in one defining relation of type $bw(x) = 1$ or $b^{-1}w(x) = 1$. This relation, as well as the relations $bb^{-1} = 1$ and $b^{-1}b = 1$, are eliminated. Both these transformations are called type-A transformations. The adjoining and the removal of the consequences of defining relations are type-B transformations.

The proof of the theorem is based on the following well-known group-theoretical theorem of Tietze.

It is possible to go over from one manner of assignment, by generating elements and defining relations, of a finitely defined group to another by using a finite number of transformations of type A and type B.

It follows from Tietze's theorem that for proving Theorem 3 it suffices to consider the case in which one of the transformations of type A or B takes place.

Thus let $\mathfrak{M}$ be a language in the alphabet $\Sigma = \{x_1, \ldots, x_n, x_1^{-1}, \ldots, x_n^{-1}\}$, that assigns the group G, and let φ be a natural homomorphism of Σ^* into G.

It is evident that the set $\mathfrak{M}$ remains unchanged in a type-B transformation. Suppose that we have a type-A transformation and that the generatrices b and b^{-1} are adjoined. To the system of defining relations we adjoin the relations $bw(x) = 1$, $bb^{-1} = 1$ and $b^{-1}b = 1$. Let $\Sigma_b = \Sigma \cup b \cup b^{-1}$ and let τ be a homomorphism of Σ_b^* onto Σ^* that is defined as follows: $\tau(x_i)=x_i$, $\tau(x_i^{-1})=x_i^{-1}$, $i=1, 2, \ldots, n$; $\tau(b)=w^{-1}(x)$, $\tau(b^{-1})=w(x)$, where $w^{-1}(x)$ is a word inverse to the word $w(x)$, i.e., a mirror image of the word $w(x^{-1})$. In the alphabet Σ_b the group G is specified by the language $\mathfrak{M}_b$, which is the kernel of the corresponding homomorphism φ_b. It is easy to see that we obtain the following commutative diagram:

$$\begin{tikzcd} \Sigma_b^* \arrow[rr, "\tau"] \arrow[dr, "\varphi_b"'] & & \Sigma^* \arrow[dl, "\varphi"] \\ & G & \end{tikzcd}$$

Hence follows that $\mathfrak{M}_b = Ker\,\varphi\tau = \tau^{-1}(\mathfrak{M})$. Since CF languages are closed under the operation of taking the pre-image, it follows that $\mathfrak{M}_b$ is likewise a CF language.

Suppose that the generatrices b and b^{-1} and the relations $bb^{-1}=1$, $b^{-1}b=1$, and $bw(x)=1$ are eliminated. Then $\Sigma=\{b, x_1, \ldots, x_n, b^{-1}, x_1^{-1}, \ldots, x_n^{-1}\}$ and let $\Sigma_b = \Sigma \setminus \{b, b^{-1}\}$.

Let us define the homomorphism $\tau: \Sigma^* \to \Sigma_b^*$, $\tau(x_i)=x_i$, $\tau(x_i^{-1})=x_i^{-1}$, $i=1, 2, \ldots, n$; $\tau(b^{-1})=w(x)$, $\tau(b)=w^{-1}(x)$. Then we have the commutative diagram

$$\begin{tikzcd} \Sigma^* \arrow[rr, "\tau"] \arrow[dr, "\varphi"'] & & \Sigma_b^* \arrow[dl, "\varphi_b"] \\ & G & \end{tikzcd}$$

In the alphabet Σ_b the group G is specified by the language $\mathfrak{M}_b = \mathrm{Ker}\,\varphi_b$. If $\mathfrak{M} = Ker\,\varphi$, then $\mathfrak{M}_b = \tau(\mathfrak{M})$. Since CF languages are closed under the operation of taking homomorphic images, it follows that $\mathfrak{M}_b$ is a CF language. This completes the proof of the theorem.

Now let us study direct products of groups specified by languages.

Let L_1 and L_2 be languages in the alphabets Σ_1 and Σ_2, and let $\Sigma_1 \cap \Sigma_2 = \varnothing$. There exists a natural homomorphism $\eta: (\Sigma_1 \cup \Sigma_2)^* \to \Sigma_1^* \times \Sigma_2^*$. We shall define the direct product of the languages L_1 and L_2 by $\eta^{-1}[(L_1, L_2)]$ and denote it by $L_1 \times L_2$; to rephrase, the direct product of the languages L_1 and L_2 is a language obtained from $L_1 \cdot L_2$ by permutations of elements of L_1 with elements of L_2.

In the following we shall need the concept of an n-store automaton. In this paper we shall use the definition given in [2].

<u>THEOREM 4.</u> Let L_1 be a language understood by the n-store automaton A_1 and L_2 a language understood by the m-store automaton A_2. Then $L_1 \times L_2$ will be understood by an n+m-store automaton A_3. If A_1 and A_2 are deterministic automata, then A_3 will also be deterministic.

<u>Proof.</u> Let $A_1 = (\mathfrak{A}_1, \Sigma_1, \delta_1, a_0, F_1, M_{11}, M_{12}, \ldots, M_{1n})$ and $A_2 = (\mathfrak{A}_2, \Sigma_2, \delta_2, b_0, F_2, M_{21}, M_{22}, \ldots, M_{2m})$, where $\mathfrak{A}_i$ is the set of internal states, Σ_i the input alphabet, δ_i the transition function, a_0 and b_0 are initial states, and F_i the set of terminal states, $i=1, 2$. M_{1j} denotes the stores of the first automaton, $j=1,2,\ldots,n$, and M_{2j} the stores of A_2, $j=1, 2, \ldots, m$. Let us construct an automaton that will be called a direct product of the automata A_1 and A_2 and denoted by $A_1 \times A_2$. $A_1 \times A_2 = (\mathfrak{A}_1 \times \mathfrak{A}_2, \Sigma_1 \cup \Sigma_2, \delta_3, (a_0, b_0), F_1 \times F_2, M_{11}, \ldots, M_{1n}, M_{21}, \ldots, M_{2m})$. The states of the automaton $A_1 \times A_2$ are elements of the direct product of sets $\mathfrak{A}_1 \times \mathfrak{A}_2$. The input alphabets are combined. The stores are combined and remain the same.

Let us assume at first that the automata A_1 and A_2 are deterministic. Then at each instant of automaton time an automaton $A_1 \times A_2$ which is in a state (a, b) can perform the following actions depending on the states a and b:

1. If A_1 in the state a reads x from the input store and thus goes over to the state a', and if A_2 in the state b reads y from the input store and goes over to b', then an automaton $A_1 \times A_2$ which is in the state (a, b) will read x from the input and go over to (a', b), or, in the case of the input y, it reads y and goes over to (a, b').

2. If a is a state belonging to the input store reading section, and if in the state b the automaton A_2 operates with internal stores (it reads or writes, or is idle), and thus goes over to the state b', then $A_1 \times A_2$ in the state (a, b) will perform the same operation with the internal store and go over into the state (a, b').

3. If A_1 in the state a operates with an internal store and thus goes over to a', then $A_1 \times A_2$ in the state (a, b) will perform the same operation with an internal store and go over to (a', b). Here the state b will have no effect whatsoever.

It is easy to see that with such an assignment of the transition function, the automaton $A_1 \times A_2$ will be a deterministic n+m-store automaton.

If A_1 and A_2 are nondeterministic automata, the transitions in the automaton $A_1 \times A_2$ will be defined in the same way, with allowance for nondeterminacy in the corresponding states. For example, the first rule is rewritten as follows:

If A_1 in the state a can read x and thus go over to a', and if A_2 in the state b can read y and go over to b', then $A_1 \times A_2$ in the state (a, b) can read x and go over to (a', b), and in the case of the input y it can read y and go over to (a, b').

Let us show that the automaton $A_1 \times A_2$ perceives the language $L_1 \times L_2$. Let $p \in (\Sigma_1 \cup \Sigma_2)^*$ and suppose that p is correctly transferring $A_1 \times A_2$ from the state (a_0, b_0) to a terminal state (a, b); p can always be expressed in the form $U_1V_1 \ldots U_tV_t$, where $U_i \in \Sigma_1^*$, $V_i \in \Sigma_2^*$, $i = 1, 2, \ldots, t$ and the internal words are not empty. From the operation of the automaton $A_1 \times A_2$ we can see that the word $p_1 = u_1u_2 \ldots u_t$ is correctly transferring the automaton A_1 from the state a_0 to the terminal state a, and the word $p_2 = v_1v_2 \ldots v_t$ is correctly transferring A_2 from the state b_0 to the terminal state b. Hence the word $p_1 \in L_1$ and $p_2 \in L_2$, and therefore $p \in L_1 \times L_2$. Now let $p \in L_1 \times L_2$. Similarly, $p = U_1V_1 \ldots U_tV_i$. Then $p_1 = U_1U_2 \ldots U_t \in L_1$ and $p_2 = V_1V_2 \ldots V_t \in L_2$; p_1 correctly transfers A_1 from the state a_0 to the terminal state a, and p_2A_2 from b_0 to the terminal state b. It is evident that in this case also the word p will correctly transfer the automaton $A_1 \times A_2$ from the state (a_0, b_0) to the terminal state (a, b). This completes the proof of the theorem.

COROLLARY 1. If L_1 is a regular language and L_2 a context-free language, then $L_1 \times L_2$ will be a context-free language.

Proof. Regular languages are understood by automata without stores, and context-free languages by 1-store automata. It follows from Theorem 4 that in this case the direct product will be represented by a 1-store automaton, i.e., it is a context-free language.

COROLLARY 2. If G_1 is a finite group and G_2 a group specified by a CF language, then $G_1 \times G_2$ will be specified by a CF language.

Proof. Let G_1 be specified by the language $\mathfrak{M}_1$, and G_2 by the language $\mathfrak{M}_2$. It is easy to see that $G_1 \times G_2$ is specified by the language $\mathfrak{M}_1 \times \mathfrak{M}_2$. If G_1 is finite, then $\mathfrak{M}_1$ will be regular and from the previous corollary it follows that $\mathfrak{M}_1 \times \mathfrak{M}_2$ is a CF language.

Let L be a language. A commutative closure of a language L is a language (denoted by A(L)) that has been obtained from L by all sorts of permutations of letters in the words occurring in L. More precisely, a word $p \in A(L)$ if and only if it can be reduced by letter permutation to a word belonging to L. The operation of commutative closure is important in the theory of context-free languages, for example in connection with Parikh's well-known theorem [5].

As before, let D_{2n} be Dick's language and let $\mathfrak{M}$ be a language containing ε. We shall use the following lemma.

LEMMA 1. $A(D_{2n}) \times \mathfrak{M}$ cannot be a context-free language for $n > 1$.

Proof. Suppose that $A(D_{2n}) \times \mathfrak{M}$ is a CF language. By a well-known criterion, there hence exist numbers p and q such that if $z \in A(D_{2n}) \times \mathfrak{M}$ and $|z| > p$, then $z = xuwvy$, $uv \neq \varepsilon$, $|uwv| < q$ and for all natural k we have $xu^kwv^ky \in A(D_{2n}) \times \mathfrak{M}$. Let a and b be distinct symbols of the alphabet of the language D_{2n}, with $a \neq b^{-1}$ and $b \neq a^{-1}$. The word $t = a^m b^m (a^{-1})^m (b^{-1})^m$ for all m belongs to $A(D_{2n}) \times \mathfrak{M}$, since $\mathfrak{M}$ contains ε. Let us take $m > p + q$. Then $t = xuwvy$ and for all k we have $xu^kwv^ky \in A(D_{2n}) \times \mathfrak{M}$. Since the alphabets of the languages $\mathfrak{M}$ and $A(D_{2n})$ are disjoint (this follows from the definition of a direct product) and $\varepsilon \in \mathfrak{M}$, we have $xu^kwv^ky \in A(D_{2n})$. The language $A(D_{2n})$ is a group language and it specifies the factor group of a free group with respect to the commutator, i.e., a free abelian group of rank n. Hence in this group we must have the relation $uwv = w$. Since $A(D_{2n})$ is a set of words whose sum of exponents of x_i is equal to the sum of exponents of x_i^{-1}, $i = 1, 2, \ldots, n$, it follows from $uwv = w$ that $uv \in A(D_{2n})$. It is evident that, if $m > q$, it is not possible to select such uv from the condition $|wv| < q$ in the word $a^m b^m a^{-m} b^{-m}$. This completes the proof of the lemma.

Let us note that it hence follows that $A(D_{2n})$ is not a CF language and that a direct product of groups specified by CF languages is not always specified by a CF language. For example, the group $Z \times Z$, where Z denotes the group of integers with respect to addition, is not specified by a CF language, since it is specified by the language $A(D_{2n})$.

Any finitely generated abelian group is isomorphic to a direct product of a finite group and a finite number of groups Z. Let us denote by Z^k the direct product of groups Z taken k times. Suppose that the abelian group G decomposes in this way into a direct product of a finite group T and Z^k.

THEOREM 5. A finitely generated abelian group $G = T \times Z^k$ will be specified by a CF language if and only if $k \leq 1$.

Proof. The group Z^k is a free abelian group of rank k and it is isomorphic to the factor group of a free group of rank k with respect to its commutator. As we have seen above, this group is specified by the language $A(D_{2k})$. A finite group T is specified by a regular language $\mathfrak{M}_T$. Then G will be specified by a language $\mathfrak{M}_T \times A(D_{2k})$, which according to the lemma cannot be context-free for $k > 1$.

On the other hand, for $k=1$ the language $A(D_2) = D_2$ will be context-free and, therefore, a CF language (by virtue of Corollary 3). This completes the proof of the theorem.

In conclusion let us examine to what extent it is essential that the alphabet Σ in which a group is defined should have the form $\Sigma = \{x_1, x_2, \ldots, x_n, x_1^{-1}, x_2^{-1}, \ldots, x_k^{-1}\}$. It is found that all the results obtained above hold for any alphabet.

Let $\Sigma_n = \{x_1, \ldots, x_n\}$. By a group language in the alphabet Σ_n we shall henceforth understand any language in the alphabet Σ_n which is a kernel of an epimorphism of Σ_n^* onto a group. Let G be a group and let φ be an epimorphism of Σ_n^* onto G. $\mathfrak{M} = \operatorname{Ker}\varphi$. We obtain the following assertion, analogous to Theorem 1:

A group G is finite if and only if $\mathfrak{M}$ is a regular language.

The proof is entirely similar to the proof of Theorem 1. It is only necessary to slightly modify the concept of a strong group automaton A_G. The input alphabet is Σ_n and the transition function δ is defined on $G \times \Sigma_n$ by the law $\delta(g, x_i) = g \circ \varphi(x_i)$.

The theory of context-free group languages is based on the following theorem, which is analogous to Theorem 3.

THEOREM 3'. Let the group G be finitely defined, $\Sigma_n = \{x_1, \ldots, x_n\}$, $\Sigma_m = \{y_1, \ldots, y_m\}$, $\varphi: \Sigma_n^* \to G$, $\psi: \Sigma_m^* \to G$, where φ and ψ are epimorphisms of Σ_n^* and Σ_m^* onto one and the same group G. Let $\mathfrak{M} = \operatorname{Ker}\varphi$ and $\mathfrak{N} = \operatorname{Ker}\psi$. Hence if $\mathfrak{M}$ is a CF language, then $\mathfrak{N}$ will also be a CF language.

Proof. Let $\Sigma_{2n} = \{x_1, x_2, \ldots, x_n, x_1^{-1}, x_2^{-1}, \ldots, x_n^{-1}\}$ and $\Sigma_{2m} = \{y_1, y_2, \ldots, y_m, y_1^{-1}, y_2^{-1}, \ldots y_m^{-1}\}$, where x_i^{-1} and y_j^{-1} are new symbols, $i = 1, 2, \ldots, n$; $j = 1, 2, \ldots, m$. Let us define the epimorphism φ': $\Sigma_{2n}^* \to G$ as follows: $\varphi'(x_i) = \varphi(x_i)$, $\varphi'(x_i^{-1}) = [\varphi(x_i)]^{-1}$. We shall also define the homomorphism $\tau: \Sigma_{2n}^* \to \Sigma_n^*$, $\tau(x_i) = x_i$, $\tau(x_i^{-1}) = p_i$, where p_i is a fixed word belonging to Σ_n^* such that $\varphi(p_i) = [\varphi(x_i)]^{-1}$. Such a word must exist, since φ is an epimorphism by our condition. It is easy to see that we obtain the following commutative diagram:

$$\begin{tikzcd} \Sigma_{2n}^* \arrow[rr, "\tau"] \arrow[dr, "\varphi'"'] & & \Sigma_n^* \arrow[dl, "\varphi"] \\ & G & \end{tikzcd}$$

Let $\mathfrak{M}' = \operatorname{Ker}\varphi'$. Then $\mathfrak{M}' = \tau^{-1}(\mathfrak{M})$ and hence $\mathfrak{M}'$ will be a CF language. The language $\mathfrak{M}'$ specifies a group G in the alphabet Σ_{2n}.

In a similar way we can define the homomorphisms ψ' and τ' and construct the commutative diagram

$$\begin{tikzcd} \Sigma_{2m}^* \arrow[rr, "\tau'"] \arrow[dr, "\psi'"'] & & \Sigma_m^* \arrow[dl, "\psi"] \\ & G & \end{tikzcd}$$

Let $\mathfrak{N} = Ker\,\psi'$. The language $\mathfrak{N}'$ specifies in the alphabet Σ_{2m} the same group G, and hence by virtue of Theorem 3 this language is context-free. Hence the language $\mathfrak{N} = \tau'(\mathfrak{N}')$ is also context-free. This completes the proof of the theorem.

It follows from this theorem that for obtaining groups specified by CF languages, it suffices to consider these groups in any convenient alphabet.

LITERATURE CITED

1. A. I. Mal'tsev, Algorithms and Recursive Functions [in Russian], Nauka, Moscow (1965).
2. V. M. Glushkov, "Simple algorithms of analysis and synthesis of store automata," Kibernetika, No. 5 (1968).
3. A. V. Anisimov, "The group of automorphisms of connected automata," in: Teoriya Avtomatov, No. 5, (1969).
4. N. Chomsky and M. P. Schützenberger, "The algebraic theory of context-free languages," in: Computer Programming and Formal Systems, North Holland, Amsterdam (1963).
5. R. J. Parikh, "Language generating devices," MIT Res. Lab. Electron. Quart. Prog. Rept., 60 (1961).