FINITE AUTOMATA OVER FREE GROUPS

JÜRGEN DASSOW

Otto-von-Guericke-Universität Magdeburg
Fakultat für Informatik PF 4120
D-39016, Magdeburg
Germany
E-mail: *dassow@iws.CS.Uni-Magdeburg.De*

VICTOR MITRANA*

Faculty of Mathematics
University of Bucharest
Str. Academiei 14
70109, Bucharest
Romania

Communicated by A. de Luca
Received 27 July 1999

AMS Mathematics Subject Classification: 68Q68, 68Q45

Finite automata are extended by adding an element of a given group to each of their configurations. An input string is accepted if and only if the neutral element of the group is associated to a final configuration reached by the automaton. We get a new characterization of the context-free languages as soon as the considered group is the binary free group. The result cannot be carried out in the deterministic case. Some remarks about finite automata over other groups are also presented.

1. Introduction

One of the oldest and most investigated machine in the automata theory is the finite automaton. Many fundamental properties have been established and many problems are still open. Unfortunately, the finite automata without any external control have a very limited accepting power. Different directions of research have been considered for overcoming this limitation.

A simple and natural extension, somehow related to the pushdown memory, was considered in a series of papers [2, 3], namely to associate to each configuration an element of a given group, but no information regarding the associated element is allowed. In [3], different types of finite automata with multiplication (the group here is the multiplicative group of nonnull rational numbers) are introduced. Paper [2] investigates finite automata endowed with a finite number of counters, each of them being able to store an integer. In our terminology, the aforementioned

*Research supported by the Alexander von Humboldt Foundation.

International Journal of Algebra and Computation Vol. 10, No. 6 (2000) 725–737

devices are finite automata over the multiplicative group of nonnull rational numbers and the additive group of integer vectors, respectively. The finite automata about which we are going to discuss in the present paper are the nondeterministic one-way finite automata endowed with one counter able to store any element of a free group.

The family of languages accepted by one-way finite automata with multiplication actually coincides with the family of unordered vector languages [1]. Languages as $\{a^n b^n | n \geq 1\}^*$ cannot be accepted by nondeterministic one-way finite automata with multiplication. In our opinion, the reason is the commutativity of the group of rational numbers with multiplication. This deadlock might be overcome if nonabelian groups are considered.

In this aim, the present paper mainly deals with finite automata over a noncommutative group, more precisely over the free group. The obtained results are considerable improvements of those in [4]. The main result is a new characterization of the context-free languages by extended finite automata over the binary free group. This result is rather unexpected since in such an automaton the same choice is available regardless the content of its counter. More precisely, the next action is determined just by the input symbol currently scanned and the state of the machine. The languages generated by regular additive valence grammars [5] are characterized by finite automata over the free group with just one generator.

2. Preliminaries

We assume the reader familiar with the basic concepts in automata and formal language theory and the group theory, free groups particularly. For further details, we refer to [6, 7], respectively.

For an alphabet V, we denote by V^* the free monoid generated by V under the operation of concatenation; the empty string is denoted by ε and $V^* - \{\varepsilon\}$ is denoted by V^+. The length of $x \in V^*$ is denoted by $|x|$, whereas $|x|_a$ is the number of occurrences of $a \in V$ in $x \in V^*$, $|x|_B$ is the number of occurrences of symbols of $B \subseteq V$ in $x \in V^*$.

Let $\mathbf{K} = (M, \circ, e)$ be a group under the operation denoted by $\circ$ with the neutral element denoted by e.

An extended finite automaton (EFA shortly) over the group $\mathbf{K} = (M, \circ, e)$ is a construct

$$A = (Q, V, \mathbf{K}, f, q_0, F)$$

where Q, V, q_0, F have the same meaning as in a usual finite automaton [6], namely the set of states, the input alphabet, the initial state and the set of final states, respectively, and $f : Q \times (V \cup \{\varepsilon\}) \longrightarrow \mathcal{P}_f(Q \times M)$.

This sort of automaton can be viewed as a finite automaton having a register in which any element of M can be written. The relation $(q, m) \in f(s, a),\ q, s \in Q,\ a \in V \cup \{\varepsilon\},\ m \in M$ means that the automaton A changes its current state s

into q, by reading the symbol a on the input tape, and writes in the register $x \circ m$, where x is the old content of the register. The initial value registered is e.

We shall use the notation

$$(q, aw, m) \models_A (s, w, m \circ r) \quad \text{iff } (s, r) \in \delta(q, a)$$

for all $s, q \in Q$ $a \in \Sigma \cup \{\varepsilon\}$, $m, r \in M$. The reflexive and transitive closure of the relation $\models_A$ is denoted by $\models_A^*$. Sometimes, the subscript identifying the automaton will be omitted when it is self-understood.

The word $x \in \Sigma^*$ is accepted by the automaton A if, and only if, there is a final state q such that $(q_0, x, e) \models^* (q, \varepsilon, e)$. In other words, a string is accepted if the automaton completely reads the string and reaches a final state with the content of the register being the neutral element of $\mathbf{K}$.

The language accepted by an extended finite automaton over a group A as above is

$$L(A) = \{x \in V^* | (q_0, x, e) \models_A^* (q, \varepsilon, e)\,, \quad \text{for some } q \in F\}\,.$$

In the following, we focus our investigation on the extended finite automata over the free groups. Note that for any (nonabelian) group $\mathbf{K}$ there is a homomorphism from a free group to $\mathbf{K}$ [7].

For a nonempty countable set M the free group generated by M is denoted by $\mathbf{F}(M)$. The free group with n generators is denoted by $\mathbf{F}_n$.

Example 2.1. Let us consider the automaton

$$(\{q, s\}, \{a, b\}, \mathbf{F}(\{\#\}), f, q, \{s\})$$

with

$$f(q, a) = \{(q, \#)\}$$

$$f(q, b) = \{(s, \#^{-1})\}$$

$$f(s, b) = \{(s, \#^{-1})\}\,.$$

Obviously, $L(A) = \{a^n b^n | n \geq 1\}$ which is not a regular language.

Remark 2.1. If we replace the state s by the state q in the above automaton, then we get an extended finite automaton recognizing the nonlinear language $\{x | x \in \{a, b\}^*, |x|_a = |x|_b\}$.

The next example is a bit more intricate.

Example 2.2. The following automaton recognizes the restricted Dyck language over the alphabet $\{a, \bar{a}\}$:

$$A = (Q, \{a, \bar{a}\}, \mathbf{F}(\{b, c\}), f, q_0, F)$$

where

$$Q = \{q_0, q_1, q_2, p_1, p_2, q_f\}$$
$$F = \{q_0, q_f\}$$

and

$$f(q_0, a) = \{(q_1, cb), (p_1, cb), (p_2, cb)\}$$
$$f(q_1, a) = \{(q_1, b), (p_1, b), (p_2, b)\}$$
$$f(q_2, a) = \{(q_1, c^{-1}b), (p_1, c^{-1}b), (p_2, c^{-1}b)\}$$
$$f(p_1, \bar{a}) = \{(q_0, b^{-1}), (p_1, b^{-1}), (p_2, b^{-1}), (q_2, b^{-1}), (q_f, b^{-1}c^{-1})\}$$
$$f(p_2, \bar{a}) = \{(q_0, c^{-1}b^{-1}), (q_f, c^{-1}b^{-1}c^{-1})\}\,.$$

Indeed, an input word is accepted if, and only if, it has the same number of occurrences of a and $\bar{a}$, respectively, and for each of its prefixes, the number of occurrences of a is at least equal with the number of occurrences of $\bar{a}$.

The family of all languages accepted by extended finite automata over the free group with n generators is denoted by $\mathcal{L}(EFA(\mathbf{F}_n))$.

We recall from [5] the definition of a grammar which generates a language in a way similar to that defined above for accepting by an extended finite automaton, namely the *additive valence grammar*. An additive valence grammar is a quintuple $G = (N, T, S, P, v)$, where (N, T, S, P) is a Chomsky generative grammar and v is a mapping from P into the set of integers. The language generated by G consists of all terminal strings w which can be derived by using the sequence $p_1, p_2, \ldots, p_n$ of rules from P with $v(p_1) + v(p_2) + \cdots + v(p_n) = 0$. Since $\mathbf{F}_1$ and the additive group of integers are isomorphic it follows that every language generated by a regular grammar with additive valences is in $\mathcal{L}(EFA(\mathbf{F}_1))$ and vice versa.

3. A New Characterization of the Context-Free Languages

Lemma 3.1. *The family of context-free languages is included in* $\mathcal{L}(EFA(\mathbf{F}_2))$.

Proof. Let L be a context-free language accepted by empty storage and final states by the pushdown automaton

$$PDA = (Q, V, \Delta, \delta, q_0, Z, F)$$

with $\Delta = \{X_1, X_2, \ldots, X_n\},\ X_1 = Z$. Moreover, without loss of generality we may assume that no transition is defined for any final state. Take $M = \{c_1, c_2, \ldots, c_n\}$ and define the mapping $\sigma : \Delta \longrightarrow \mathbf{F}(M)$ as $\sigma(X_i) = c_i,\ 1 \leq n$. Construct the extended finite automaton over the free group $\mathbf{F}(M)$

$$A = (Q \cup \{s_0\}, V, \mathbf{F}(M), f, s_0, F)$$

where

$$f(s_0, \varepsilon) = \{(q_0, c_1)\}$$

and

$$f(q,a) = \bigcup_{X\in\Delta} \{(p, (\sigma(X))^{-1}\sigma(Y_m)\cdots\sigma(Y_2)\sigma(Y_1)) | (p, Y_1Y_2\cdots Y_m) \in \delta(q,a,X)\}$$

$$\cup \bigcup_{X\in\Delta} \{(p, (\sigma(X))^{-1} | (p,\varepsilon) \in \delta(q,a,X)\}$$

for all $a \in V \cup \{\varepsilon\}$ and $q \in Q$. One can easily shown, by induction on n, that

$$(q, xy, R_1R_2\cdots R_m) \models^n_{PDA} (q', y, R'_1R'_2\cdots R'_s)$$

if and only if

$$(q, xy, \sigma(R_m)\sigma(R_{m-1})\cdots\sigma(R_1)) \models^n_A (q', y, \sigma(R'_s)\sigma(R'_{s-1})\cdots\sigma(R'_1)).$$

Thus, at every step the two automata are in the same state, and the content of the register of A becomes the neutral element of $\mathbf{F}(M)$ exactly when the pushdown memory becomes empty. By the initial supposition regarding PDA we conclude that $L(A) = L(PDA)$ holds.

Since each free group is a subgroup of the binary free group, the proof is complete. □

Lemma 3.2. *The family $\mathcal{L}(EFA(\mathbf{F}_2))$ is included in the family of context-free languages.*

Proof. Let L be a language accepted by the extended finite automaton

$$A = (Q, V, \mathbf{F}(M), f, q_0, F)$$

where $M = \{c_1, c_2\}$. Take $V_2 = \{a_i, \bar{a}_i | i = 1, 2\}$ such that $V_2 \cap V = \emptyset$, and define the mapping $h : \mathbf{F}(M) \longrightarrow V_2^*$ as follows:

$$h(e) = \varepsilon,\ h(c_i) = a_i,\ h(c_i^{-1}) = \bar{a}_i, i = 1, 2$$

$$h(x) = h(y_1)h(y_2)\cdots h(y_m)\,, \quad \text{where } y_1y_2\cdots_m \text{ is the shortest element of } \mathbf{F}(M)$$
$$\text{with } x \equiv y_1y_2\cdots y_m,\ y_i \in M \cup M^{-1}, 1 \le i \le m\,.$$

We construct the right-linear grammar

$$G = (Q, V \cup V_2, q_0, P)$$

with the following set of rules P:

$$P = \{q \longrightarrow ah(r)p | (p, r) \in f(q, a)\}$$
$$\cup \{q \longrightarrow ah(r) | (p, r) \in f(q, a), p \in F\}$$
$$\cup \{q \longrightarrow h(r)p | (p, r) \in f(q, \varepsilon)\}$$
$$\cup \{q \longrightarrow h(r) | (p, r) \in f(q, \varepsilon), p \in F\}\,.$$

We claim that

$$L(A) = g((D_2 \sqcup\!\sqcup V^*) \cap L(G))$$

where

- g is a homomorphism from $(V_2 \cup V)^*$ into V^* which erases all symbols in V_2 and leaves unchanged all symbols in V.
- $\sqcup\!\sqcup$ is the shuffle operation defined recursively by

$$(u \sqcup\!\sqcup \varepsilon) = (\varepsilon \sqcup\!\sqcup u) = \{u\}$$

  and

$$(au \sqcup\!\sqcup bv) = a(u \sqcup\!\sqcup bv) \cup b(au \sqcup\!\sqcup v)\,,$$

  where u, v are strings and a, b are symbols, and extended in a natural way to languages:

$$L_1 \sqcup\!\sqcup L_2 = \bigcup_{u \in L_1, v \in L_2} (u \sqcup\!\sqcup v)$$

- D_2 is the Dyck language over V_2 generated by the context-free grammar having the following set of productions:

$$R = \{S \longrightarrow SS, S \longrightarrow \varepsilon\} \cup \{S \longrightarrow a_i S \bar{a}_i, S \longrightarrow \bar{a}_i S a_i | i = 1, 2\}\,.$$

Indeed, the strings in $D_2 \sqcup\!\sqcup V^*$ can be reduced to the empty string by removing all letters from V and then by removing iteratively all substrings $a_1\bar{a}_1$, $a_2\bar{a}_2$, $\bar{a}_1 a_1$, $\bar{a}_2 a_2$, while the strings in $L(G)$ contain the symbols from V in the order in which the automaton A accepts them. More precisely, all words obtained from those of $L(G)$ by removing the symbols not in V are accepted by A, but not necessarily with the neutral element in its register. By intersection, one gets words that have to satisfy both aforementioned conditions. The homomorphism g erases all letters not in V, which concludes the proof of our claim.

It is known that $L \sqcup\!\sqcup R$ is a context-free language providing that L is a context-free language and R is a regular language. From the well-known properties of closure of the family of context-free languages it follows that $L(A)$ is a context-free language. □

Denote by REG, CF, the families of regular and context-free languages, respectively.

Theorem 3.1. *$REG = \mathcal{L}(EFA(\mathbf{F}_0)) \subset \mathcal{L}(EFA(\mathbf{F}_1)) \subset \mathcal{L}(EFA(\mathbf{F}_2)) = CF$.*

Proof. The equality $REG = \mathcal{L}(EFA(\mathbf{F}_0))$ is obvious, whereas the relation $CF = \mathcal{L}(EFA(\mathbf{F}_2))$ is a direct consequence of the previous lemmas.

From Example 2.1, it follows that $REG \subset \mathcal{L}(EFA(\mathbf{F}_1))$. Remember that $\mathcal{L}(EFA(\mathbf{F}_1))$ equals the family of languages generated by regular additive valence grammars. But the language $L = \{a^n b^n a^m b^m | n, m \geq 1\}$ does not belong to this family [5]. Actually, the reader can easily infer a complete reasoning quite similar to the proof of the fact that the linear language $\{a^n b^m a^m b^n | n, m \geq 1\}$ cannot be accepted by any extended automaton over the additive group of integers, which is given in the proof of Lemma 4.3.

On the other hand, the language L is accepted by the following extended automaton:

$$A = (\{q_0, q_1, q_2, q_3\}, \{a, b\}, \mathbf{F}(\{\$, \#\}), f, q_0, \{q_3\})$$

where the transition mapping f is defined by

$$f(q_0, a) = \{(q_0, \#)\}, \qquad f(q_0, b) = \{(q_1, \#^{-1})\}$$

$$f(q_1, a) = \{(q_2, \$)\}, \qquad f(q_1, b) = \{(q_1, \#^{-1})\}$$

$$f(q_2, a) = \{(q_2, \$)\}, \qquad f(q_2, b) = \{(q_3, \$^{-1})\}$$

$$f(q_3, b) = \{(q_3, \$^{-1})\}.$$

Therefore, $\mathcal{L}(EFA(\mathbf{F}_1)) \subset \mathcal{L}(EFA(\mathbf{F}_2))$ which completes the proof. □

4. Deterministic EFA over Free Groups

Denote by $\mathcal{L}(DEFA(\mathbf{F}_n))$ the family of languages recognized by deterministic EFA over the group $\mathbf{F}_n$.

Lemma 4.1. *The languages*

$$L_1 = \{a^n | n \geq 1\} \cup \{a^n b^n | n \geq 1\}$$

$$L_2 = \{a, b\}^* \backslash \{a^n b^n | n \geq 1\}$$

$$L_3 = \{a^n b^m | n \geq m \geq 1\}$$

are not in $\mathcal{L}(DEFA(\mathbf{F}_n))$, *for any positive integer* n.

Proof. Let $A = (Q, \{a, b\}, \mathbf{F}(M), \delta, q_0, F)$ be a DEFA over $\mathbf{F}(M)$, for some set M, such that $L_1 = L(A)$. Since $a^n \in L(A)$, for all $n \geq 1$, there are the integers $1 \leq r < s \leq card(F) + 1$ such that $(q_0, a^r, e) \models_A^* (q, \varepsilon, e)$ and $(q_0, a^s, e) \models_A^* (q, \varepsilon, e)$, for some $q \in Q$. Since $a^n b^n \in L(A)$, for all $n \geq 1$, we have also $(q_0, a^r b^r, e) \models_A^* (q, b^r, e) \models_A^* (q', \varepsilon, e)$, for some $q' \in F$. Hence, $(q_0, a^s b^r, e) \models_A^* (q, b^r, e) \models_A^* (q', \varepsilon, e)$, implying $a^s b^r \in L(A)$, contradiction. In conclusion, $L_1 \neq L(A)$. In a similar way one can prove that L_2 and L_3 do not belong to $\mathcal{L}(DEFA(\mathbf{F}_n))$, for any positive integer n. □

Lemma 4.2. *For all* $n \geq 1$, *the strict inclusion* $\mathcal{L}(DEFA(\mathbf{F}_n)) \subset \mathcal{L}(EFA(\mathbf{F}_n))$ *holds.*

Proof. Let $A = (\{q_0, q_1, q_2\}, \{a, b\}, \mathbf{F}(\{m\}), \delta, q_0, \{q_1, q_2\})$ be a deterministic EFA whose transition mapping δ is defined as follows:

$$\delta(q, a) = \begin{cases} \{(q_1, e), (q_2, m)\} & \text{if } q = q_0 \\ \{(q_1, e)\} & \text{if } q = q_1 \\ \{(q_2, m)\} & \text{if } q = q_2 \end{cases}$$

$$\delta(q_2, b) = \{(q_2, m^{-1})\}\,.$$

It is easy to observe that $L(A) = \{a^n | n \geq 1\} \cup \{a^n b^n | n \geq 1\}$. □

Lemma 4.3. *The family of linear languages is incomparable with each of the families $\mathcal{L}(DEFA(\mathbf{F}_1))$, $\mathcal{L}(DEFA(\mathbf{F}_2))$, $\mathcal{L}(EFA(\mathbf{F}_1))$.*

Proof. There are nonlinear languages in $\mathcal{L}(DEFA(\mathbf{F}_1))$ (see Remark 2.1) and the language L_1 from Lemma 4.1 is linear. This shows the incomparability relation between the family of linear languages and $\mathcal{L}(DEFA(\mathbf{F}_1))$ and $\mathcal{L}(DEFA(\mathbf{F}_2))$, respectively.

On the other hand, the linear language $L = \{a^n b^m a^m b^n | n, m \geq 1\}$ cannot be accepted by any extended finite automaton over the additive group of integers. Let us suppose that $L = L(A)$, $A = (Q, \{a, b\}, \mathbf{Z}, f, q_0, F)$. All strings in L, excepting a finite number of them, are accepted by A following derivations of the form:

$$\begin{aligned}(q_0, a^n b^m a^m b^n, 0) &\models^* (q_1, a^{i_1} b^m a^m b^n, k_1) \models^* (q_1, a^{i_2} b^m a^m b^n, k_2) \\ &\models^* (q_2, b^{j_1} a^m b^n, k_3) \models^* (q_2, b^{j_2} a^m b^n, k_4) \\ &\models^* (q_3, a^{j_3} b^n, k_5) \models^* (q_3, a^{j_4} b^n, k_6) \\ &\models^* (q_4, b^{i_3}, k_7) \models^* (q_4, b^{i_4}, k_8) \models^* (q, \varepsilon, 0)\end{aligned}$$

for some $q \in F$.

Clearly, we must have

$$k_{2t} - k_{2t-1} \neq 0\,, \quad t = 1, 2, 3, 4$$

$$(k_2 - k_1)(k_8 - k_7) < 0 \quad \text{and} \quad (k_4 - k_3)(k_6 - k_5) < 0\ .$$

Assume, without loss of generality, that $k_2 - k_1 > 0$ and $k_4 - k_3 > 0$. Because L is infinite, we may assume that the derivation $(q_1 a^{i_1} b^m a^m b^n, k_1) \Longrightarrow (q_1 a^{i_2} b^m a^m b^n, k_2)$ is used t times, $t - k_4 + k_3 > 0$, and the derivation $(q_2 b^{j_1} a^m b^n, k_3) \Longrightarrow (q_2 b^{j_2} a^m b^n, k_4)$ is used r times. Due to the equality

$$t(k_2 - k_1) + r(k_4 - k_3) = (t - k_4 + k_3)(k_2 - k_1) + (r + k_2 - k_1)(k_4 - k_3)$$

it follows that the string

$$a^{n-(k_4-k_3)(k_2-k_1)} b^{m+(k_4-k_3)(k_2-k_1)} a^m b^n$$

is also accepted by the automaton A, which is a contradiction. □

Lemma 4.4. *$\mathcal{L}(DEFA(\mathbf{F}_n))$ is strictly contained in the family of deterministic context-free languages.*

Proof. Let $A = (Q, V, \mathbf{F}(M), \delta, q_0, F)$ be an EFA over the free group generated by M. It suffices to consider a set M with just two elements, namely $M = \{X, Y\}$. We define the mapping $\sigma : \mathbf{F}(M) \longrightarrow \{X, Y, X', Y'\}^*$ as follows:

$$\sigma(e) = \varepsilon,\ \sigma(Z) = Z,\ \sigma(Z^{-1}) = Z' \quad \text{for } Z \in \{X, Y\},$$

$$\sigma(x) = \sigma(y_1)\sigma(y_2)\cdots\sigma(y_m)\,, \quad \text{where } y_1y_2\cdots_m \text{ is the shortest element of } \mathbf{F}(\mathrm{M})$$
$$\text{with } x \equiv y_1y_2\cdots y_m,\ y_i \in M \cup M^{-1}, 1 \le i \le m\,,$$

and construct the deterministic pushdown automaton

$$B = (Q', V, \{X, Y, X', Y', \$\}, \theta, q_0, \$, F')$$

where

$$Q' = Q \cup \{q_f | q \in F\} \cup \{\langle p, x\rangle | p \in Q,\ x \in \{X, Y, X', Y'\}^*,\ |x| \le k_A\}$$

with $k_A = \max\{|\sigma(m)| : \delta(q, a) = (q', m),\ \text{for some } q, q' \in Q,\ a \in V \cup \{\varepsilon\}\}$, the set of final states $F' = \{q_f | q \in F\}$, and the transition function θ defined in the following way:

(1) $\theta(q, a, Z) = (\langle p, \sigma(m)\rangle, Z),\ q \in Q \cup F',\ Z \in \{X, Y, X', Y', \$\},\ \delta(q, a) = (p, m)$
(2) $\theta(\langle p, Zx\rangle, \varepsilon, Z') = (\langle p, x\rangle, \varepsilon),\ Z \in \{X, Y\}$
(3) $\theta(\langle p, Z'x\rangle, \varepsilon, Z) = (\langle p, x\rangle, \varepsilon), Z \in \{X, Y\}$
(4) $\theta(\langle p, Zx\rangle, \varepsilon, U) = (p, x^R ZU)$, for all Z, U, excepting the cases aforementioned
(5) $\theta(\langle p, \varepsilon\rangle, \varepsilon, Z) = \begin{cases} (p, Z),\ Z \in \{X, Y, X', Y'\} \\ (p_f, \$),\ Z = \$\,. \end{cases}$

Note that x^R denotes the mirror image of the string x.

As one can easily see the pushdown automaton above is deterministic. We state that

$$(q, xy, m) \models_A^* (p, y, m') \quad \text{iff } (q, xy, \sigma(m)^R\$) \models_B^* (p, y, \sigma(m')^R\$)\,.$$

Each transition in the automaton A is simulated in the automaton B as follows:

- firstly, one applies the transition (1)
- one applies as many times as possible the transitions (2) and (3)
- the transitions (4) and (5) are applied once successively.

Thus the "only if" part merely follows. A similar argument is valid for the "if" part as well. Indeed, let us emphasize all steps in an arbitrary computation of B where the current state is from $Q \cup F'$. Let

$$(q_1, x_1, \alpha_1) \models_B^* (q_2, x_2, \alpha_2) \models_B^* \cdots \models_B^* (q_r, x_r, \alpha_r)$$

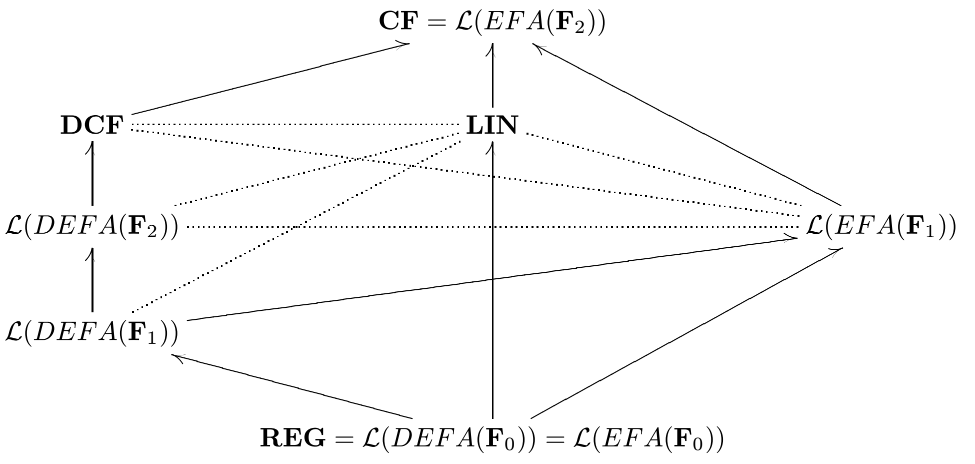

Fig. 1. $X \to Y$ indicates a strict inclusion $X \subset Y$, and the families connected by a dotted line are incomparable.

be such a computation. Every part of this computation is just a step in a computation in A. From the above reasoning, the equality $L(A) = L(B)$ follows. The fact that $\{a^n | n \geq 1\} \cup \{a^n b^n | n \geq 1\}$ is a deterministic context-free language completes the proof. $\square$

The results obtained so far are summarized by the next theorem where LIN and DCF denote the class of linear languages and deterministic context-free languages, respectively.

Theorem 4.1. *Figure* 1 *holds.*

Theorem 4.2. *The families* $\mathcal{L}(DEFA(\mathbf{F}_n))$, $n \geq 1$, *are closed under intersection with regular languages and inverse homomorphisms and not closed under complement, union, letter-to-letter homomorphisms, concatenation.*

Proof. Let $A = (Q, \Sigma, \mathbf{F}_n, \delta, q_0, F)$ be a deterministic EFA over $\mathbf{F}_n$, for some $n \geq 1$, and let $B = (K, \Sigma, \theta, s_0, F')$ be a deterministic finite automaton. We construct the deterministic EFA $C = (Q \times K, \Sigma, \mathbf{F}_n, \varphi, (q_0, s_0), F \times F')$ with the transition relation

$$\varphi((q,s),a) = \{(q', \theta(s,a)), m)\}\,, \quad \text{where } a \in \Sigma \text{ and } (q', m) = \delta(q, a)$$

$$\varphi((q,s),\varepsilon) = \{(z', y), m)\}\,, \quad \text{where } (z', m) = \delta(z, a)\,.$$

Obviously, $((q_0, s_0), w, e) \models_C^* ((q,s), \varepsilon, m)$ iff $(q_0, w, e) \models_A^* (q, \varepsilon, m)$ and $(s_0, w) \models_B^* (s, \varepsilon)$ which proves the closure under intersection with regular sets.

To show the closure under inverse homomorphism, let $A = (Q, \Sigma, \mathbf{F}_n, \delta, q_0, F,)$ be a DEFA, and let $h : \Delta^* \to \Sigma^*$ be a homomorphism. We construct the DEFA $B = (Q', \Delta, \mathbf{F}_n, \varphi, q_0, F)$, where the set of states is defined by

$$Q' = Q \cup \{[w_1, q, w_2] | q \in Q, w_1 w_2 = h(a) \quad \text{for all } a \in \Delta\}\,.$$

The transition function φ is defined by

$$\varphi(q,a) = ([\varepsilon, q, h(a)], e),\ q \in Q, a \in \Delta$$

$$\varphi([w_1, q, uw_2], \varepsilon) = ([w_1 u, q', w_2], m) | (q', m) \in \delta(q, u), u \in \Sigma \cup \{\varepsilon\}, uw_2 \in \Sigma^+\}$$

$$\varphi([w, q, \varepsilon], \varepsilon) = \begin{cases} ([w, q', \varepsilon], m) & \text{if } (q', m) = \delta(q, \varepsilon) \\ (q, e) & \text{if } \delta(q, \varepsilon) \text{ is not defined} \end{cases}.$$

Note that $(q, a, e) \models_C^* (q', \varepsilon, m)$, for $q, q' \in Q, a \in \Delta \cup \{\varepsilon\}$ iff $(q, h(a), e) \models_A^* (q', \varepsilon, m)$, i.e. a computation of C on the input word $w \in \Delta^*$ simulates a computation of A on the input $h(w) \in \Sigma^*$.

The languages

$$K_1 = \{a^n | n \geq 1\}$$

$$K_2 = \{a^n b^n | n \geq 1\}$$

$$K_3 = \{c^n | n \geq 1\} \cup \{a^n b^n | n \geq 1\}$$

are in $\mathcal{L}(DEFA(\mathbf{F}_1))$. With the notations of Lemma 4.1, we obtain the following relations:

- L_2 is the complement of K_2
- $L_1 = K_1 \cup K_2$
- $L_3 = K_1 K_2$
- $L_1 = h(K_3)$ with the homomorphism $h : \{a, b, c\}^* \to \{a, b\}^*$, $h(a) = h(c) = a$, $h(b) = b$.

By Lemma 4.1, the nonclosure of $\mathcal{L}(DEFA(\mathbf{F}_n))$ under complement, union, product and homomorphisms follows. □

5. Connections with Other Groups

In this section, we shall provide some connections with other groups, especially with the groups of integers and rational numbers with the addition and the multiplication operations, respectively.

Denote by $\mathcal{L}(EFA(M, \circ, e))$ the family of languages accepted by extended finite automata over the group $(M, \circ, e)$. Clearly, we have

- $\mathcal{L}(EFA(\mathbf{Q} - \{0\}, \cdot, 1))$ is the family of languages accepted by nondeterministic one-way finite automata with multiplication [3].
- $\mathcal{L}(EFA(\mathbf{Z}, +, 0))$ is the family of languages generated by regular grammars with additive valences [5].

Note that $\mathcal{L}(EFA(\mathbf{Z}, +, 0)) = \mathcal{L}(EFA(\mathbf{F}_1))$.

Obviously, if two groups $(M_1, \perp, e_1)$ and $M_2, \top, e_2)$ are isomorphic, then $\mathcal{L}(EFA(M_1, \perp, e_1)) = \mathcal{L}(EFA(M_2, \top, e_2))$. Naturally, one may ask: does the reciprocal assertion hold? Somewhat surprisingly, the answer is negative.

Theorem 5.1. $\mathcal{L}(EFA(\mathbf{Z}, +, 0)) = \mathcal{L}(EFA(\mathbf{Q}, +, 0))$.

Proof. It suffices to prove only the inclusion $\mathcal{L}(EFA(\mathbf{Q}, +, 0)) \subseteq \mathcal{L}(EFA(\mathbf{Z}, +, 0))$. Take an extended finite automata over $(\mathbf{Q}, +, 0)$

$$A = (K, V, (\mathbf{Q}, +, 0), f, q_0, F)$$

and denote by

$$R_A = \left\{\frac{m}{n} \Big| \text{ exist } q, q' \in K, a \in V \cup \{\varepsilon\} \text{ such that } \left(q', \frac{m}{n}\right) \in f(q, a)\right\}$$

$$\pi_A = \prod_{\frac{m}{n} \in R_A} n\,.$$

Construct the extended finite automata over the additive group of integers

$$A' = (K, V, (\mathbf{Z}, +, 0), f', q_0, F)$$

with

$$f'(q, a) = \left\{\left(q', \frac{m}{n}\pi_A\right) \Big| \left(q', \frac{m}{n}\right) \in f(q, a)\right\}$$

for all $q, q' \in K$ and $a \in V \cup \{\varepsilon\}$.

The equality $L(A) = L(A')$ follows, due to the next relation

$$\sum_{\frac{m}{n} \in R_A} \alpha_{(\frac{m}{n})} \frac{m}{n} = 0 \quad \text{iff } \pi_A \sum_{\frac{m}{n} \in R_A} \alpha_{(\frac{m}{n})} \frac{m}{n} = 0$$

for all natural numbers $\alpha_{(\frac{m}{n})}$, $\frac{m}{n} \in R_A$. □

On the other hand, in [1], it was proved that the family of languages generated by regular additive valence grammars is strictly included in the family of unordered vector languages. Consequently, we have

Theorem 5.2. $\mathcal{L}(EFA(\mathbf{Q}, +, 0)) \subset \mathcal{L}(EFA(\mathbf{Q} - \{0\}, \cdot, 1))$.

References

1. J. Dassow and Gh. Păun, *Regulated rewriting in formal language theory*, Akademie-Verlag, Berlin, 1989.
2. S. A. Greibach, *Remarks on blind and partially blind one-way multicounter machines*, Theor. Comp. Sci. **7** (1978), 311–324.
3. O. H. Ibarra, S. K. Sahni and C. E. Kim, *Finite automata with multiplication*, Theor. Comp. Sci. **2** (1976), 271–294.
4. V. Mitrana, *Valence grammars on a free generated group*, Bull. EATCS **47** (l992), 174–179.

5. Gh. Păun, *A new generative device: valence grammars*, Rev. Roum. Math. Pures et Appl. **25** (6) (1980), 911–924.
6. G. Rozenberg and A. Salomaa, *Handbook of Formal Languages*, Springer-Verlag, Berlin, Vol. 1, 1997.
7. J. J. Rotman, *An Introduction to the Theory of Groups*, Springer-Verlag, 1995.