Semigroup Forum (2008) 76: 159–168
DOI 10.1007/s00233-007-9024-6

RESEARCH ARTICLE

Two-letter group codes that preserve aperiodicity of inverse finite automata

Jean-Camille Birget · Stuart W. Margolis

Communicated by Benjamin Steinberg

Received 9 January 2007 and in final form 10 August 2007 / Published online: 18 October 2007
© Springer Science+Business Media, LLC 2007

Abstract We construct group codes over two letters (i.e., bases of subgroups of a two-generated free group) with special properties. Such group codes can be used for reducing algorithmic problems over large alphabets to algorithmic problems over a two-letter alphabet. Our group codes preserve aperiodicity of inverse finite automata. As an application we show that the following problems are PSPACE-complete for *two-letter* alphabets (this was previously known for large enough finite alphabets): The intersection-emptiness problem for inverse finite automata, the aperiodicity problem for inverse finite automata, and the closure-under-radical problem for finitely generated subgroups of a free group. The membership problem for 3-generated inverse monoids is PSPACE-complete.

Keywords Free groups · Inverse semigroups · Inverse automata

1 Introduction

Codes and coding theory are a well-known and important subject. In its most general form, a code over an alphabet A is defined to be a subset C of A^* such that any concatenation of elements of C can be uniquely factored, or "decoded", into a

Both authors were supported in part by NSF grant DMS-9970471. The first author was also supported in part by NSF grant CCR-0310793. The second author acknowledges the support of the Excellency Center, "Group Theoretic Methods for the Study of Algebraic Varieties" of the Israeli Science Foundation.

J.-C. Birget
Dept. of Computer Science, Rutgers University at Camden, Camden, NJ 08102, USA
e-mail: birget@camden.rutgers.edu

S.W. Margolis (✉)
Dept. of Mathematics, Bar Ilan University, Ramat Gan 52900, Israel
e-mail: margolis@macs.biu.ac.il

sequence of elements of C. Equivalently, the submonoid $\langle C\rangle$ of A^* generated by C is free with base C, i.e., $\langle C\rangle$ is isomorphic to the free monoid C^*. As a reference see [5]. Some notation: A^* denotes the free monoid over A, i.e., the set of all finite sequences ("words") of elements of A, including the empty word. A^+ denotes the free semigroup over A, i.e., the set of all non-empty finite sequences over A.

For groups one can use the same definition of a code, replacing "free monoid" by "free group". In the literature such a code is called a *base of a free group*. We'll call it *group code* because we will use it in the spirit of information coding. A precise definition of a group code appears below. First we need some notation: The free group over a generating set A is denoted by $\mathrm{FG}(A)$. We use a copy $A^{-1} = \{a^{-1} : a \in A\}$ of A, disjoint from A, to denote the inverses of the generators. We denote $A \cup A^{-1}$ by $A^{\pm 1}$. For $w = a_1 \dots a_n$ with $a_1, \dots, a_n \in A^{\pm 1}$, the inverse of w is defined to be $w^{-1} = a_n^{-1} \dots a_1^{-1}$, where $(a^{-1})^{-1}$ is always replaced by a for all $a \in A$. The identity element of $\mathrm{FG}(A)$ is the empty word, and is denoted by 1. The elements of $\mathrm{FG}(A)$ are all the words over the alphabet $A^{\pm 1}$ that are *reduced*, i.e., that contain no subsegment of the form $a\,a^{-1}$ or $a^{-1}a$ (for any $a \in A$). In general, for any word $w \in (A^{\pm 1})^*$ we define the *reduction of* w to be the word obtained by cancelling all subsegments of the form $w\,w^{-1}$ (with $w \in (A^{\pm 1})^*$) iteratively as much as possible, and we denote the resulting reduced word by $\mathrm{red}(w)$. For any word w we denote its length by $|w|$. See [8, 11, 12] for background on free groups.

Any function $f : A \to (B^{\pm 1})^*$ can be extended (uniquely) to a group morphism $f^{(G)} : \mathrm{FG}(A) \to \mathrm{FG}(B)$ defined by $f^{(G)}(a_1^{\varepsilon_1} \dots a_n^{\varepsilon_n}) = \mathrm{red}(f(a_1)^{\varepsilon_1} \dots f(a_n)^{\varepsilon_n})$, where $\varepsilon_1, \dots, \varepsilon_n \in \{-1, 1\}$.

Important convention: Throughout this paper we will view the free group $\mathrm{FG}(A)$ as a subset of the free monoid $(A^{\pm 1})^*$; namely, $\mathrm{FG}(A)$ consists of all the reduced words over $A^{\pm 1}$. Of course, $\mathrm{FG}(A)$ is only a subset of $(A^{\pm 1})^*$, not a submonoid.

Definition 1.1 Let $\varphi : A \to (B^{\pm 1})^*$ be a map whose extension to a free-group morphism $\varphi^{(G)} : \mathrm{FG}(A) \to \mathrm{FG}(B)$ is *injective*. Then the image set $\varphi^{(G)}(A)$ ($\subset \mathrm{FG}(B) \subset (B^{\pm 1})^*$) is called a *group code* over B, and the elements of $\varphi^{(G)}(A)$ are called *code words*. By our convention, $\mathrm{FG}(B)$ is a subset of $(B^{\pm 1})^*$, and hence a group code is a set of words.

The injective map $\varphi^{(G)}|_A : A \to \mathrm{FG}(B)$ defined by $a \mapsto \mathrm{red}(\varphi(a))$, i.e., the restriction of $\varphi^{(G)}$ to A, is called a *group encoding* of A over B.

The study of free groups and of bases of free groups (i.e., group codes) has a long history [8, 11, 12]. In particular, Nielsen showed in the 1920s that every finitely generated subgroup of a free group is itself free and hence has a group code. A little later in the 1920s Schreier extended Nielsen's result to all subgroups of a free group. So, group codes can be finite or infinite. We note the following however:

Proposition 1.2 *An infinite group code cannot be a regular language, but can be deterministic context-free.*

Proof If an infinite regular group code existed we could apply the Pumping Lemma, so the group code would contain all words of the form $w_n = ux^nv$ ($n \in \mathbb{N}$), for some fixed words u, x, v, with x non-empty. But then the following non-trivial relation would hold among code words: $w_2 w_1^{-1} w_2 = w_3$.

The example $\{a^n b a^{-n} : n \geq 0\}$ over the alphabet $\{a, b\}^{\pm 1}$, shows that there are infinite group codes that are deterministic context-free languages. The set $\{a^n b a^{-n} : n \geq 0\}$ is a well-known Nielsen basis. □

We are interested in group codes over an alphabet of size 2. Just as for the usual codes (over monoids), the main purpose of group codes is to translate large alphabets into smaller alphabets. This in turn can be used to show that some problems that are hard over large alphabets are also hard over a 2-letter alphabet. We will consider the fixed two-letter alphabet $\{a, b\}$ and the inverses a^{-1}, b^{-1} of these letters.

Subgroups of a free group are closely related to inverse monoids and inverse finite automata [13]. By definition, an *inverse finite automaton* is a structure $\mathcal{A} = (Q, X, \delta, q_0, q_f)$ where, according to the standard notation in [9], Q is the set of states, q_0 is the start state, and q_f is the accept state. For inverse automata, the input alphabet is $X \cup X^{-1} = X^{\pm 1}$, although we only mention X explicitly; the designation "inverse" automatically provides the inverse letters. The state-transition relation δ is a partial function $\delta : Q \times X^{\pm 1} \to Q$, and is required to have the following property: For each letter $x \in X$, the partial function $\delta(\cdot, x) : q \in Q \mapsto \delta(q, x) \in Q$ is *injective*. Moreover, we require that the partial function $\delta(\cdot, x^{-1})$ be the inverse of $\delta(\cdot, x)$. We represent an inverse finite automaton by its state-graph, in the same way as for ordinary finite automata (see [9]), except that we omit the edges labeled by inverse letters. More precisely, when $\delta(p, x) = q$ (with $p, q \in Q$, $x \in X$) we draw an edge $p \xrightarrow{x} q$; we implicitly also have an edge $q \xrightarrow{x^{-1}} p$, but we don't draw that edge. See e.g. [6] for more information on inverse automata.

Let $\kappa : X^{\pm 1} \to (\{a, b\}^{\pm 1})^*$ be any group encoding and let $\mathcal{A}$ be any inverse finite automaton $\mathcal{A}$ with input alphabet X. We define the *encoded inverse finite automaton* $\kappa(\mathcal{A})$, with input alphabet $\{a, b\}$, by the following two-step construction:

(1) We replace every edge $p \xrightarrow{x} q$ of $\mathcal{A}$ (with $x \in X$) by a path labeled by $\kappa(x)$; to do this we introduce $|\kappa(x)| - 1$ new states and $|\kappa(x)|$ new edges. Implicitly, we now also have the inverses of the new edges, thus obtaining a path from q to p labeled by $\kappa(x^{-1})$. Let $\kappa(\mathcal{A})_0$ be the nondeterministic finite automaton obtained so far.

(2) Starting from $\kappa(\mathcal{A})_0$ we apply the *fold* operation as much as possible. This means that any two edges (explicitly drawn or implicit) with a common beginning or end vertex, and with identical label in $\{a, b\}^{\pm 1}$ are made equal. For example, if $p \xrightarrow{x^e} q_1$ and $p \xrightarrow{x^e} q_2$ are present (with $e \in \{-1, 1\}$) then one folding step makes q_1 equal to q_2, and the above two edges become equal. See e.g., [6, 13, 14] for more information on the very classical fold operation. In particular, it is well known that maximal folding produces a unique resulting automaton, which does not depend on the folding sequence chosen. We denote this resulting automaton by $\kappa(\mathcal{A})$; it is an inverse automaton if $\mathcal{A}$ is an inverse automaton. We denote the transition function of $\kappa(\mathcal{A})$ by δ_κ.

In general, for any automaton $\mathcal{M}$ we let $L_{\mathcal{M}}$ denote the language accepted by $\mathcal{M}$. For an inverse automaton $\mathcal{A} = (Q, A, \delta, q_0, q_f)$ we consider the language accepted $L_{\mathcal{A}} \subseteq (A^{\pm 1})^*$, as well as the group language of $\mathcal{A}$, defined as follows:

Definition 1.3 The *group language* of a finite inverse automaton $\mathcal{A}$ with input alphabet A consists of the *reduced* words ($\in (A^{\pm 1})^*$) accepted by $\mathcal{A}$; in other words, the group language of $\mathcal{A}$ is $L_{\mathcal{A}} \cap \mathrm{FG}(A)$.

Lemma 1.4 *For a finite inverse automaton $\mathcal{A}$ with input alphabet A the group language $L_{\mathcal{A}} \cap \mathrm{FG}(A) = \mathrm{red}(L_{\mathcal{A}})$.*

Proof This is Lemma 1.1 in [6]. □

Note that by Benois' theorem [3, 4], $\mathrm{red}(L_{\mathcal{A}})$ is also accepted by a finite automaton with alphabet $A^{\pm 1}$. But this automaton cannot be an inverse automaton, except in trivial cases. Indeed, an inverse automaton will always accept some non-reduced words (except when $L_{\mathcal{A}}$ is empty or consists of just the empty word).

An *automaton with involution* over the alphabet $(A^{\pm 1})^*$ is an automaton $\mathcal{A}$ such that for every edge $p \xrightarrow{x} q$ with $x \in (A^{\pm 1})^*$, of $\mathcal{A}$, $q \xrightarrow{x^{-1}} p$ is also an edge of $\mathcal{A}$. We will always assume that all automata over the alphabet $A^{\pm 1}$ are automata with involution. Notice that an automaton with involution is deterministic if and only if it is an inverse automaton.

Let $\mathcal{A}$ be any automaton with involution over the alphabet $A^{\pm 1}$. The folded automaton $\rho(\mathcal{A})$ is defined as above by applying some maximal folding sequence to $\mathcal{A}$. This determines an equivalence relation $\sim$ on the states of $\mathcal{A}$ by defining two states to be equivalent if they define the same state of $\rho(\mathcal{A})$, that is, if the two states are folded onto one another. Recall that a Dyck word over $(A^{\pm 1})^*$ is a word that reduces to the identity word in $\mathrm{FG}(A)$. The language of Dyck words is known to be the smallest language containing the empty word and closed under concatenation and the conjugation operation $w \mapsto awa^{-1}$, for all $a \in A^{\pm 1}$.

Lemma 1.5 *Let $\mathcal{A}$ be an automaton with involution over the alphabet $(A^{\pm 1})$. Then states p, q of $\mathcal{A}$ satisfy $p \sim q$ if and only if there is a Dyck word w such that w labels a path from p to q in $\mathcal{A}$.*

Proof Assume that the reduced automaton $\rho(\mathcal{A})$ is obtained by a sequence of m foldings. Let $\mathcal{A}_i$ be the automaton obtained after i foldings, $0 \le i \le m$. There is a corresponding equivalence relation $\sim_i$ on the states of $\mathcal{A}$, and $\sim_0 \subset \sim_1 \subset \cdots \subset \sim_m = \sim$.

We will prove by induction that if i is the least integer such that $p \sim_i q$, then there is a Dyck word w that labels a path from p to q in $\mathcal{A}$. This is true if $i = 0$ since then the empty word labels a path from p to itself.

Assume that if $r \sim_i s$ then there is a Dyck word labeling a path from r to s in $\mathcal{A}$; and assume that $p \sim_{i+1} q$, but $p \not\sim_i q$. Since a folding identifies exactly two states, the $(i+1)$st folding identifies the $\sim_i$ class of p with that of q. Let $[r]_{\sim_i}$ denote the $\sim_i$ equivalence class of a state r of $\mathcal{A}$.

Thus there is a $\sim_i$ equivalence class, X, such that there are edges of $\mathcal{A}_i$, $[p]_{\sim_i} \xrightarrow{x} X$ and $X \xleftarrow{x} [q]_{\sim_i}$ for some $x \in A^{\pm 1}$. It is clear that every path in $\mathcal{A}_i$ lifts, by "unfolding", to a path of $\mathcal{A}$. Thus in $\mathcal{A}$ there are states p', q' and states $r, s \in X$ such that $p' \in [p]_{\sim_i}$, $q' \in [q]_{\sim_i}$ and $p' \xrightarrow{x} r$ and $s \xleftarrow{x} q'$ in $\mathcal{A}$. Since $p \sim_i p' \xrightarrow{x}$

$r \sim_i s \xleftarrow{x} q \sim_i q'$ we have, by induction, Dyck words u, v, w that label paths from p to p', q' to q and r to s respectively in $\mathcal{A}$. Therefore the Dyck word $uxwx^{-1}v$ labels a path from p to q in $\mathcal{A}$.

Conversely, a straightforward induction on the length of a Dyck word w shows that if w labels a path from a state p to a state q of $\mathcal{A}$ then $p \sim q$. □

Corollary 1.6 *Let $\mathcal{A}$ be an automaton with involution over the alphabet $A^{\pm 1}$ and let $\rho(\mathcal{A})$ be the reduced automaton of $\mathcal{A}$. Let p, q be states of $\mathcal{A}$. If $w = a_1 \ldots a_n$, with $a_i \in A^{\pm 1}$, $1 \le i \le n$, labels a path from $[p]_\sim$ to $[q]_\sim$ in $\rho(\mathcal{A})$, then there are Dyck words $u_0, \ldots, u_n$ such that $u_0 a_1 \ldots a_n u_n$ labels a path from p to q in $\mathcal{A}$. In particular,* $\mathrm{red}(L(\mathcal{A})) = \mathrm{red}(L(\rho(\mathcal{A})))$.

Proof There are states $p = p_0, p_1, \ldots, p_n = q$ of $\mathcal{A}$ such that $[p_i]_\sim \xrightarrow{a_{i+1}} [p_{i+1}]_\sim$ are edges of $\rho(\mathcal{A})$. Since paths in $\rho(\mathcal{A})$ lift to paths of $\mathcal{A}$, there are states $p'_0, p'_1, \ldots, p'_n$ of $\mathcal{A}$ such that $p_i \sim p'_i$ for $0 \le i \le n$, and such that there are edges $p'_i \xrightarrow{a_{i+1}} p'_{i+1}$ of $\mathcal{A}$. By Lemma 1.5, there are Dyck words $u_0, \ldots, u_n$ such that $p_i \xrightarrow{u_i} p'_i$ and the first assertion of the corollary follows.

It is clear that $\mathrm{red}(L(\mathcal{A})) \subseteq \mathrm{red}(L(\rho(\mathcal{A})))$ since paths in $\mathcal{A}$ fold to paths in $\rho(\mathcal{A})$. The converse inclusion follows from the first assertion of the corollary if we take w to be a reduced word. □

We record a special case of the above corollary that is of special interest in this paper in the proposition below.

Proposition 1.7 *Let $\kappa : X \to (A^{\pm 1})^*$ be any group encoding, and let $\kappa^{(M)} : (X^{\pm 1})^* \to (A^{\pm 1})^*$ be the corresponding monoid morphism. Let $\mathcal{A}$ be an inverse finite automaton with alphabet X and let $L_{\mathcal{A}} \subseteq (X^{\pm 1})^*$ be the language it accepts. Then the group language of $\kappa(\mathcal{A})$ is* $\mathrm{red}(\kappa^{(M)}(L_{\mathcal{A}}))$. *In other words,* $\mathrm{red}(L_{\kappa(\mathcal{A})}) = \mathrm{red}(\kappa^{(M)}(L_{\mathcal{A}}))$.

2 Aperiodicity preserving group codes

Some standard definitions: A monoid M is called *aperiodic* iff $x^{n+1} = x^n$ for all $x \in M$, for some constant n depending only on M. A finite automaton $\mathcal{A}$ is called aperiodic iff the syntactic monoid of $\mathcal{A}$ is aperiodic.

Let Y be a finite subset of $\mathrm{FG}(A)$, and let $H = \langle Y \rangle$ be the subgroup of $\mathrm{FG}(A)$ generated by Y. Then we can construct a finite inverse automaton $\mathcal{A}_H$ with the following property: A reduced word $w \in \mathrm{FG}(A)$ belongs to $H = \langle Y \rangle$ iff $\mathcal{A}_H$ accepts w. In other words: The group language $L(\mathcal{A}_H) \cap \mathrm{FG}(A)$ of $\mathcal{A}_H$ is H. A construction of $\mathcal{A}_H$ goes as follows (see [6], p. 251, for more details): Consider cyclic graphs labeled by the elements of Y, and glue these cycles together at their origins; if we now pick this common origin as the start and accept state we obtain a nondeterministic automaton. Next, we apply maximal folding. The resulting finite inverse automaton is $\mathcal{A}_H$. One can show that it only depends on H (not on the originally given generating set Y).

Definition 2.1 A subgroup H of a group G is *closed under radical* (also called "radical-closed", or "pure") iff for all $g \in G$ and all $N > 0$ we have: $g^N \in H$ implies $g \in H$.

The radical of H in G is the set $\sqrt{H} = \{g \in G : \text{there exists } N > 0 \text{ with } g^N \in H\}$.

Closure under radical for subgroups of a free group is intimately connected to aperiodicity of inverse automata:

Lemma 2.2 *Let Y be a finite subset of* $\mathrm{FG}(A)$. *The subgroup $H = \langle Y \rangle$ of* $\mathrm{FG}(A)$ *generated by Y is closed under radical iff the finite inverse automaton $\mathcal{A}_H$ is aperiodic.*

Proof This is Theorem 3.1 in [6]. □

Proposition 2.3 (Transitivity of radical closure) *Consider subgroups $K \leq H \leq G$ such that K is radical-closed in H and H is radical-closed in G. Then K is radical-closed in G.*

Proof Suppose $x \in G$ is such that $x^n \in K$, for some integer $n \geq 2$. Then $x^n \in H$, hence $x \in H$, by radical closure of H in G. So we have now $x \in H$ and $x^n \in K$. This implies that $x \in K$, by radical closure of K in H. □

Definition 2.4 A group homomorphism $h : \mathrm{FG}(X) \to \mathrm{FG}(A)$ preserves closure under radical iff for every subgroup H of $\mathrm{FG}(X)$ we have: H is closed under radical in $\mathrm{FG}(X)$ iff $h(H)$ is closed under radical in $\mathrm{FG}(A)$.

A group encoding $\varphi : X \to (A^{\pm 1})^*$ is said to preserve closure under radical iff the group homomorphism $\varphi^{(G)} : \mathrm{FG}(X) \to \mathrm{FG}(A)$ determined by φ preserves closure under radical.

Proposition 2.5 *Let $f : \mathrm{FG}(X) \to \mathrm{FG}(A)$ be an injective morphism such that the image group* $\mathrm{Im}(f)$ *of f is radical-closed in* $\mathrm{FG}(A)$. *Then for all subgroups H of* $\mathrm{FG}(X)$ *we have*: *H is radical-closed in* $\mathrm{FG}(X)$ *iff $f(H)$ is radical-closed in* $\mathrm{FG}(A)$. *In other words*:

A group encoding φ preserves radical-closure iff $\mathrm{Im}(\varphi)$ (*reduced in the free group*) *is radical-closed.*

Proof Suppose $f(H)$ is radical-closed in $\mathrm{FG}(A)$. Then $f(H)$ is also radical-closed in $\mathrm{Im}(f)$. Hence, since f is an isomorphism between the groups $\mathrm{FG}(X)$ and $\mathrm{Im}(f)$, H is radical-closed in $\mathrm{FG}(X)$.

Suppose H is radical-closed in $\mathrm{FG}(X)$. Then $f(H)$ is radical-closed in $\mathrm{Im}(f)$, since f is an isomorphism between $\mathrm{FG}(X)$ and $\mathrm{Im}(f)$. Hence, since $\mathrm{Im}(f)$ is radical-closed in $\mathrm{FG}(A)$, transitivity of radical closure implies that $f(H)$ is also radical-closed in $\mathrm{FG}(A)$. □

Example (A family of finite aperiodic two-letter group codes of all sizes) Consider the finite set $C_n = \{a^i b a^{-i} : 0 \leq i \leq n-1\}$, over the alphabet $\{a, b\}^{\pm 1}$. It is well known that this set has the Nielsen property, hence it is a group code (compare with

Ex. 3, Sect. 3.2, p. 138 in [12]). Moreover, consider the inverse automaton $\mathcal{A}$ given by the following transition table (with state set $\{1, 2, \ldots, n\}$, with 1 as both start and accept state):

	1	2	...	$n-1$	n
a	2	3	...	n	$-$
b	1	2	...	$n-1$	n

An easy calculation then shows that

$$\mathrm{red}(L_{\mathcal{A}}) = \mathrm{red}(\langle C_n \rangle),$$

where "red" refers to reduction in $\mathrm{FG}(\{a, b\})$. In other words, the free group $\mathrm{red}(\langle C_n \rangle)$ is the group language of $\mathcal{A}$.

The syntactic inverse monoid of $\mathcal{A}$ is generated by the identity map, corresponding to the letter b, and the partial map $i \in \{1, 2, \ldots, n-1\} \longmapsto i+1$ (undefined on n), corresponding to the letter a. One verifies immediately that every element in this inverse monoid is either the empty map or a (partial) map of the form $\varphi_{h,k,j} : x \in \{h, \ldots, k\} \longmapsto x + j$ (undefined for $x \notin \{h, \ldots, k\}$), for some $1 \le h \le k \le n$ and $-n+1 \le j \le n-1$. When $j = 0$, $\varphi_{h,k,j}$ is a (partial) identity, hence an idempotent. When $j \neq 0$, $(\varphi_{h,k,j})^n$ is the empty map. In any case, $(\varphi_{h,k,j})^{n+1} = (\varphi_{h,k,j})^n$, hence the monoid is *aperiodic*.

In summary we have:

Proposition 2.6 *For any alphabet $X = \{x_1, x_2, \ldots, x_n\}$ of size n, the map $f : x_i \mapsto a^{i-1} b a^{-i+1}$ $(1 \le i \le n)$ is a group encoding into a two-generated free group that preserves closure under radical.*

By combining the above lemmas and propositions we obtain:

Corollary 2.7 *Let f be the group encoding defined in Proposition 2.6. Let $\{w_1, \ldots, w_k\}$ be any finite set of words $\subset (X^{\pm 1})^*$. Then the subgroup $\langle w_1, \ldots, w_k \rangle$ of* $\mathrm{FG}(X)$ *is closed under radical iff the subgroup $\langle f(w_1), \ldots, f(w_k) \rangle$ of* $\mathrm{FG}(\{a, b\})$ *is closed under radical.*

Application: complexity of radical-closure and aperiodicity

Group encodings are log-space computable reductions from large alphabets to small alphabets. This enables us to show that the problems below about inverse finite automata or about free groups are PSPACE-complete over two-letter alphabets. Previously it was known that they are PSPACE-complete over all large enough finite alphabets ([6], Theorem 6.13).

The *aperiodicity problem* takes as input a finite automaton and asks whether the language accepted by this automaton is aperiodic. S. Cho and D. Huynh [7] showed that the aperiodicity problem for general finite automata is PSPACE-complete, and it was shown in [6] (Theorem 6.13) that the problem remains PSPACE-complete for inverse finite automata (over some fixed finite alphabet).

The *radical-closure problem* for a free group $\mathrm{FG}(X)$ takes as input a list of words $w_1, \ldots, w_n \in \mathrm{FG}(X)$, and asks whether the subgroup $\langle w_1, \ldots, w_n \rangle$ of $\mathrm{FG}(X)$ generated by these words is closed under radical. It was proved in [6] (Theorem 7.1) that this problem is PSPACE-complete for some fixed finite alphabet X. We can now strengthen these results:

Theorem 2.8 *The radical-closure problem for a free group with* two generators, *and the aperiodicity problem for inverse finite automata over a* two-letter *alphabet*, *are* PSPACE-*complete*.

Proof By Corollary 2.7, the group encoding f is a reduction of the radical-closure problem over any fixed finite alphabet to the radical-closure problem over a two-letter alphabet. It was shown in [6] (Theorem 3.6) that the radical-closure problem and the aperiodicity of inverse finite automata are polynomial-time reducible to each other; in this reduction, the alphabets are preserved.

Finally, as we saw above, the radical-closure problem is PSPACE-complete over some finite alphabet, and is in PSPACE for all finite alphabets. □

3 Other applications of group codes

As we saw, a group encoding is a log-space computable function from a possibly large alphabet problems to a possibly small alphabet. This will enables us to show that the problems below about inverse finite automata or about free groups are PSPACE-complete over a two- or three-letter alphabet.

The *intersection-emptiness problem* for finite automata takes as input a list of finite automata $\mathcal{A}_i$ $(i = 1, \ldots, k)$ where k is part of the input, and asks whether the intersection of the languages accepted by these automata is empty. For general deterministic finite automata this problem was shown to be PSPACE-complete by D. Kozen [10], and for inverse finite automata PSPACE-completeness was shown in [6] (Proposition 5.3).

Theorem 3.1 *The intersection-emptiness problem for inverse finite automata over a fixed* two-letter *alphabet is* PSPACE-*complete*.

Proof Let $\mathcal{A}_1, \ldots, \mathcal{A}_n$ be inverse finite automata with alphabet A and let $L_1, \ldots, L_n \subseteq (A^{\pm 1})^*$ be the respective languages that they accept. Let $f : A \to (B^{\pm 1})^*$ be any group encoding with $|B| = 2$, and let $L'_1, \ldots, L'_n \subseteq (B^{\pm 1})^*$ be the languages accepted by the inverse finite automata $f(\mathcal{A}_1), \ldots, f(\mathcal{A}_n)$ respectively.

We claim that $L_1 \cap \cdots \cap L_n = \varnothing$ iff $L'_1 \cap \cdots \cap L'_n = \varnothing$, which shows that f reduces the intersection emptiness problem of inverse automata over the alphabet A to the intersection emptiness problem of inverse automata over the alphabet B.

If $L_1 \cap \cdots \cap L_n \neq \varnothing$ consider $w \in L_1 \cap \cdots \cap L_n$. By Lemma 1.4 we can assume that w is reduced. Then, by Proposition 1.7, $\mathrm{red}(f(w)) \in L'_1 \cap \cdots \cap L'_n$; hence $L'_1 \cap \cdots \cap L'_n \neq \varnothing$.

Conversely, if $y \in L'_1 \cap \cdots \cap L'_n$ ($\neq \varnothing$) we can again assume by Lemma 1.4 that y is reduced. Then by Proposition 1.7, $y \in \mathrm{red}(f(L_1)) \cap \cdots \cap \mathrm{red}(f(L_n))$. Since the function $F = \mathrm{red}(f(\cdot)) : \mathrm{FG}(A) \to \mathrm{FG}(B)$ is injective (by definition of a group code), it has an inverse function F^{-1} and we have $F^{-1}(y) \in L_1 \cap \cdots \cap L_n$. So, $L_1 \cap \cdots \cap L_n \neq \varnothing$.

Finally, as we saw above, the intersection-emptiness problem is PSPACE-complete over some finite alphabet. So the reduction makes the encoded problems PSPACE-complete over a two-letter alphabet. □

The *membership problem* for finite inverse monoids is defined as follows: The input is a finite list of injective partial maps $f_0, f_1, \dots, f_m$ on a finite set $\{1, \dots, n\}$. Each f_i is described by a function table that bijectively maps a subset of $\{1, \dots, n\}$ to a subset of $\{1, \dots, n\}$; entries in the table where f_i is not defined are blank. The question is whether f_0 can be written as a composition of some of the f_i and f_i^{-1} (for $1 \le i \le m$); more rigorously, the question is whether f_0 belongs to the inverse monoid generated by $\{f_1, \dots, f_m\}$. Below we will also consider the membership problem for 3-generator finite inverse monoids; here the input consists of four injective partial maps f_0, f_1, f_2, f_3, and the question is the same as before (now with $m = 3$).

PSPACE-completeness of the membership problem for general functions was shown by D. Kozen [10]. For permutations the problem is in the complexity class NC (hence in P), as proved by L. Babai, E. Luks, A. Seress [1]. In [2] M. Beaudry, P. McKenzie, D. Thérien proved that the membership problem for general functions (not assumed to be injective) remains PSPACE-complete if the monoid generated by $\{f_1, \dots, f_m\}$ is assumed to be in certain pseudo-varieties, and is NP-complete or in NP or in P for certain other pseudo-varieties.

Although inverse monoids are similar to groups in many ways, problems about inverse monoids can be much harder than the corresponding problems about groups:

Theorem 3.2 *The membership problem for the class of finite* inverse *monoids is* PSPACE-complete. *The problem remains* PSPACE-complete *if the finite inverse monoids are required to have just three generators.*

Proof Since we showed that the intersection-emptiness problem is PSPACE-complete for inverse finite automata with a two-letter input alphabet, we can apply Kozen's reduction (see p. 263 of [10]). Kozen's proof needs a few changes in order to make his functions injective.

Let $\mathcal{A}_i = (Q_i, \Sigma, \delta_i, q_i^{(\mathrm{start})}, q_i^{(\mathrm{fin})})$ (for $i = 1, \dots, k$) be a sequence of inverse finite automata, with the same two-letter alphabet $\Sigma = \{\alpha, \beta\}$. We can assume that $q_i^{(\mathrm{start})} \neq q_i^{(\mathrm{fin})}$ (see [6]). As the set acted on by our partial functions we take $S = \{o_1, o_2\} \cup \bigcup_{i=1}^{k} Q_i$. The (partial) functions are defined as follows:

For each $a \in \Sigma$, define $f_a : S \to S$ by $f_a(q_i) = \delta_i(q_i, a)$ (for $q_i \in Q_i$), and $f_a(o_2) = o_2$; however, $f_a(o_1)$ is undefined. Also, consider the function $f_{\mathrm{init}} : S \to S$ defined by $f_{\mathrm{init}}(q_i^{(\mathrm{start})}) = q_i^{(\mathrm{start})}$ for $i = 1, \dots, k$, and $f_{\mathrm{init}}(o_1) = o_2$, and f_{init} is undefined elsewhere. Finally, the "test function" $f_0 : S \to S$ is defined by $f_0(q_i^{(\mathrm{start})}) = q_i^{(\mathrm{fin})}$ for $i = 1, \dots, k$, and $f_0(o_1) = o_2$, and f_0 is undefined elsewhere.

Now it is straightforward to check (exactly as in [10], p. 263) that f_0 is generated by $\{f_{\mathrm{init}}, f_\alpha, f_\beta\}^{\pm 1}$ iff $\bigcap_{i=1}^{k} L_{\mathcal{A}_i} \neq \varnothing$. □

References

1. Babai, L., Luks, E., Seress, A.: Permutation groups in NC. In: Proc. 19th ACM Annual Symposium on Theory of Computing, pp. 409–420 (1987)
2. Beaudry, M., McKenzie, P., Thérien, D.: The membership problem in aperiodic transformation monoids. J. ACM **39**(3), 599–616 (1992)
3. Benois, M.: Parties rationnelles du groupe libre. C. R. Acad. Sci. Paris Sér. A **269**, 1188–1190 (1969)
4. Berstel, J.: Transductions and Context-Free Languages. Teubner, Stuttgart (1979)
5. Berstel, J., Perrin, D.: Theory of Codes. Academic, New York (1985)
6. Birget, J.C., Margolis, S., Meakin, J., Weil, P.: PSPACE-complete problems for subgroups of free groups. Theor. Comput. Sci. **242**, 247–281 (2000)
7. Cho, S., Huynh, D.: Finite-automaton aperiodicity is PSPACE-complete. Theor. Comput. Sci. **88**, 99–116 (1991)
8. Cohen, D.E.: Combinatorial Group Theory: A Topological Approach. London Math. Soc. Student Texts, vol. 14. Cambridge University Press, Cambridge (1989)
9. Hopcroft, J., Ullman, J.: Introduction to Automata Theory, Languages, and Computation. Addison-Wesley, Reading (1979) (original edition)
10. Kozen, D.: Lower bounds for natural proof systems. In: Proc. 18th IEEE Annual Symposium on Foundations of Computer Science, pp. 254–266 (1977)
11. Lyndon, R., Schupp, P.: Combinatorial Group Theory. Springer, New York (1977)
12. Magnus, W., Karrass, A., Solitar, D.: Combinatorial Group Theory. Dover, New York (1976), Interscience, New York (1966)
13. Margolis, S., Meakin, J.: Free inverse monoids and graph immersions. Int. J. Algebra Comput. **3**, 79–100 (1993)
14. Stallings, J.: The topology of graphs. Invent. Math. **71**, 551–565 (1983)