Computing relative abelian kernels of finite monoids

Edite Cordeiro and Manuel Delgado*

Abstract

Let H be a pseudovariety of abelian groups corresponding to a recursive supernatural number. In this note we explain how a concrete implementation of an algorithm to compute the kernel of a finite monoid relative to H can be achieved. The case of the pseudovariety Ab of all finite abelian groups was already treated by the second author and plays an important role here, were we will be interested in the proper subpseudovarieties of Ab. Our work relies on an algorithm obtained by Steinberg.

Introduction and motivation

The problem of computing kernels of finite monoids goes back to the early seventies and became popular among semigroup theorists through the Rhodes Type II conjecture which proposed an algorithm to compute the kernel of a finite monoid relative to the class G of all finite groups. Proofs of the conjecture were given in independent and deep works by Ash [1] and Ribes and Zalesskiĭ [14]. For an excellent survey on the work done around this conjecture, as well as connections with other topics such as the Mal'cev product, we refer the reader to [12].

The work of Ribes and Zalesskiĭ solves a problem on profinite groups (the product of a finite number of finitely generated subgroups of a free group is closed for the profinite topology of the free group) which in turn, using work of Pin and Reutenauer [13], solves the Type II conjecture. Pin and Reutenauer essentially reduced the problem of determining the kernel of a finite monoid to the problem of determining the closure of a finitely generated subgroup of a free group endowed with the profinite topology. This idea was followed by several authors to compute kernels relative to other classes of groups, considering in these cases relatively free groups endowed with topologies given by the classes in cause. We can refer Ribes and Zalesskiĭ [15] for the class of all finite p-groups, the second author [3] for the class Ab of all finite abelian groups, and Steinberg [17] for any class of finite abelian groups closed under the formation of homomorphic images, subgroups and finite direct products. A class of finite groups closed under the formation of homomorphic images, subgroups and finite direct products is called a *pseudovariety of groups*.

*The authors gratefully acknowledge support of *Fundação para a Ciência e Tecnologia* through the *Centro de Matemática da Universidade do Porto*.

Steinberg's paper [17] gives an algorithm, on which this work is based, to compute the kernel of a finite monoid relative to any pseudovariety of abelian groups. Since the problem of the existence of an algorithm for the case of locally finite pseudovarieties (which are pseudovarieties containing the free object in the variety they generate) is trivial, and Steinberg's paper was mostly dedicated to theoretical results, it emphasizes the cases of non locally finite pseudovarieties. We are aiming to obtain concrete implementations and therefore even the locally finite case requires some work. Concrete implementations of this kind of algorithms are useful, since calculations (unduable by hand due to the time required) often give the necessary intuition to formulate conjectures and may help in the subsequent problem solving. A step towards the concrete implementation in the GAP system [18] for the case of the pseudovariety Ab was given in [4] by the second author who also implemented it using the GAP programming language. This algorithm is presently part of a GAP package which will probably also contain implementations of the algorithms described in this paper. The usefulness of this software can be inferred from a number of papers whose original motivation came from computations done: we can refer several joint works by Fernandes and the second author [5, 6, 7, 8].

In the first section of the present paper we recall a few facts concerning the concept of supernatural number and mention a bijective correspondence between the classes of supernatural numbers and pseudovarieties of abelian groups.

In the second section we observe that computing the closure of a subgroup of $\mathbb{Z}^n$ (relative to certain topologies) is feasible without too much work using GAP. Notice that we are aiming to use Steinberg's algorithm to compute relative kernels which, as already observed, uses computing relative closures as an essential ingredient.

The third section is dedicated to the computation of the closure of semilinear sets relative to the profinite topology. It is relevant for Section 4.

In the fourth section we recall the definition of kernel of a finite monoid relative to a pseudovariety of groups. Then we dedicate two subsections to the correction of the concrete implementations we are proposing. The cases of pseudovarieties of abelian groups corresponding to infinite supernatural numbers and those of pseudovarieties corresponding to natural numbers are treated separately.

Applications will appear in forthcoming papers by the authors and V.H. Fernandes.

1 Supernatural numbers and pseudovarieties of abelian groups

A finite abelian group G is, via the fundamental theorem of finitely generated abelian groups, isomorphic to a product $\mathbb{Z}/m_1\mathbb{Z} \times \cdots \times \mathbb{Z}/m_r\mathbb{Z}$ of cyclic groups, where the m_i $(1 \leq i \leq r)$ are positive integers such that $m_i \mid m_{i+1}$, $1 \leq i \leq r-1$. The m_i's are known as the *torsion coefficients* of G.

A *supernatural number* is a formal product of the form Πp^{n_p} where p runs over all positive prime numbers and $0 \leq n_p \leq +\infty$. We say that a supernatural number Πp^{n_p} has *finite* *support* if all n_p, except possibly a finite number, are zero. A supernatural number of finite support is said to be *finite* if all n_p are finite. We sometimes refer to the finite supernatural numbers as *natural numbers*, since the correspondence is obvious. The set of natural numbers is denoted by $\mathbb{N}$. The other supernatural numbers are said to be *infinite*. There are evident notions of *greatest common divisor* (*gcd*) and *least common multiple* (*lcm*) of supernatural numbers generalizing the corresponding notions for natural numbers. For example, $gcd(2^2 \times 3, 2^{+\infty}) = 2^2 = 4$, and $lcm(2^2 \times 3, 2^{+\infty}) = 2^{+\infty} \times 3$.

To a supernatural number π one can associate the pseudovariety H_π of all finite abelian groups whose torsion coefficients divide π, that is, the pseudovariety generated by the cyclic groups $\{\mathbb{Z}/n\mathbb{Z} : n \mid \pi\}$. For example, to 2^∞ one associates the pseudovariety of all 2-groups which are abelian; to the natural number 2 one associates the pseudovariety generated by the cyclic group $\mathbb{Z}/2\mathbb{Z}$ and to the supernatural number Πp^∞, where p runs over all positive prime numbers, is associated the pseudovariety Ab of all finite abelian groups. Conversely, to a pseudovariety H of abelian groups one can associate the supernatural number $\pi_\mathsf{H} = lcm(\{n : \mathbb{Z}/n\mathbb{Z} \in \mathsf{H}\})$. We thus have a bijective correspondence between pseudovarieties of abelian groups and supernatural numbers. This correspondence is in fact a lattice isomorphism [17].

A supernatural number is said to be *recursive* if the set of all natural numbers which divide it is recursive. In particular, supernatural numbers of finite support are recursive.

2 Relative closures of subgroups of the free abelian group

For a pseudovariety H of groups and a finite set A, we denote by $F_\mathsf{H}(A)$ the relatively free group on A in the variety of groups (in the Birkhoff sense) generated by H.

Proposition 2.1 *[17] Let π be a supernatural number and let A be a set of cardinality $n \in \mathbb{N}$. Then if $\pi \in \mathbb{N}$, $F_{\mathsf{H}_\pi}(A) = (\mathbb{Z}/\pi\mathbb{Z})^n$. Otherwise, i.e. when π is infinite, $F_{\mathsf{H}_\pi}(A) = \mathbb{Z}^n$, the free abelian group on n generators.*

It turns out that the pseudovarieties of abelian groups corresponding to natural numbers are locally finite, while those corresponding to infinite supernatural numbers are not locally finite. The relatively free groups appearing in the last proposition will be turned into topological spaces, the finite ones being discrete. In the remaining part of this section we will be interested in computing the closure of subgroups of these relatively free groups, thus the only non trivial case occurs when π is an infinite supernatural number. We assume that π is infinite for the rest of this section. The relatively free group in cause is then the free abelian group $\mathbb{Z}^n$ itself, but it will be endowed with a topology that depends on π. The pro-H_π topology on $\mathbb{Z}^n$ is the least topology rendering continuous all homomorphisms into groups of H_π. This topology may be described in other ways; for example, one can take as a basis of neighborhoods of the neutral element all subgroups N such that $\mathbb{Z}^n/N \in \mathsf{H}_\pi$ and then make $\mathbb{Z}^n$ a topological group in the standard way. The pro-Ab topology of an abelian group G is in general called simply *profinite* topology of G.

The following result, when π is recursive, gives an algorithm to compute the pro-H_π closure of a subgroup of $\mathbb{Z}^n$. For a subset X of $\mathbb{Z}^n$, we denote by $\mathsf{Cl}_{\mathsf{H}_\pi}(X)$ the pro-H_π closure of X.

Proposition 2.2 *[17] Let $\{e_1, \ldots, e_n\}$ be a basis of $\mathbb{Z}^n$ and let π be an infinite supernatural number. Let $a_1, \ldots, a_k$ be positive integers and consider the subgroup $G = \langle a_1e_1, \ldots, a_ke_k\rangle$. For each i, let $b_i = gcd(a_i, \pi)$. Then $\mathsf{Cl}_{\mathsf{H}_\pi}(G) = \langle b_1e_1, \ldots, b_ke_k\rangle$.*

We explain next how we can make use of Proposition 2.2 to compute in practice the pro-H_π closure of a given subgroup of $\mathbb{Z}^n$.

For some theory concerning the notions that follow, which involve, in particular, the use of normal forms of matrices to represent abelian groups, see, for instance, [2, 16]. A subgroup G of $\mathbb{Z}^n$ can be specified by giving a $n \times n$ matrix B whose rows (some of which may consist entirely of zeros) generate G. We then have: $G = \langle B\rangle = \{uB : u \in \mathbb{Z}^n\}$. In particular, a basis of $\mathbb{Z}^n$ can be specified by an invertible $n \times n$ integer matrix, that is, an integer matrix with determinant ± 1. The set of such matrices is denoted by $\mathrm{GL}(n, \mathbb{Z})$.

Let G be a subgroup of $\mathbb{Z}^n$. There exists a basis $\{e_1, \ldots, e_n\}$ of $\mathbb{Z}^n$ such that the set $\{a_1e_1, \ldots, a_ke_k\}$, where $a_1 \mid a_2 \mid \cdots \mid a_k$, is a basis of G. This statement, which in general appears as part of the proof of the fundamental theorem of finitely generated abelian groups, could thus be written as follows: there exists a matrix $C \in \mathrm{GL}(n, \mathbb{Z})$ and a matrix S in Smith Normal Form (the one whose non-zero entries are the a_i's) such that SC represents a basis of G.

Suppose that G is given through a matrix B representing it. Next we explain how the matrix C representing a basis of $\mathbb{Z}^n$ as well as the matrix S referred above can be computed. Using GAP [18] one can efficiently compute invertible integer matrices P and Q such that $PBQ = S$ where S is in Smith Normal Form. Then Q^{-1} is the matrix representing the basis of $\mathbb{Z}^n$ we are looking for. To verify this, it suffices to note that the rows of $PB = SQ^{-1}$ generate G. The a_i's are the non-zero entries of S.

Let π be an infinite recursive supernatural number and let S and Q^{-1} be as in the preceding paragraph. Denote by $\overline{S}$ the matrix obtained from S by replacing each a_i by $b_i = gcd(a_i, \pi)$. Then, using Proposition 2.2, we get $\mathsf{Cl}_{\mathsf{H}_\pi}(G) = \langle \overline{S}Q^{-1}\rangle = \{u\overline{S}Q^{-1} : u \in \mathbb{Z}^n\}$. Note that assuming that π is recursive, $gcd(a_i, \pi)$ is computable. Moreover, if we assume that π is of finite support, the computation of $gcd(a_i, \pi)$ can be carried out without difficulties.

3 The profinite closure of a semilinear set

Let M be a finite monoid generated by n elements. There exists a finite ordered set A of cardinality n and a surjective homomorphism $\varphi : A^* \to M$ from the free monoid on A onto M. From now on we consider φ fixed, which means that we fix the A-generated monoid M. We fix also the canonical homomorphism $\gamma : A^* \to \mathbb{Z}^n$ defined by $\gamma(a_i) = (0, \ldots, 0, 1, 0, \ldots, 0)$ (1 is in the position i) where a_i is the i^{th} element of A. For $w \in A^*$, the i^{th} component of $\gamma(w)$ is the number of occurrences of the i^{th} letter of A in w. Given a set $X \subseteq A^*$ we refer $\gamma(X)$ as the *commutative image* of X.

As we will see in Section 4, to be able to compute $\mathsf{Cl}_{\mathsf{Ab}}(\gamma(\varphi^{-1}(x)))$ is essential for the implementations of the algorithms to compute relative abelian kernels we are proposing.

Let $x \in M$. A natural way to compute (a regular expression for) $\varphi^{-1}(x)$ is to consider the automaton $\Gamma(M, x)$ obtained from the right Cayley graph of M by taking the neutral element as the initial state and x as final state. Notice that the language of $\Gamma(M, x)$ is precisely $\varphi^{-1}(x)$. One could expect that the commutative image $\gamma(\varphi^{-1}(x))$ would then be easily computed, but in practice this is not the case, as we explain next. A regular expression for the regular language $\varphi^{-1}(x)$ is obtained by handling an automaton whose number of states is precisely the number of elements of M. It is known (see, for instance, [4]) that the size of a regular expression for the language of an automaton grows exponentially with the number of states of the automaton, thus we may obtain a huge expression for $\varphi^{-1}(x)$. We must recall that γ has then to be applied, in some way, to this expression, and this may be impracticable.

An approach for the computation of an expression of relatively small size for the language recognized by an automaton is discussed in [11]. The implementation in the GAP package [10] of an algorithm to compute such an expression from a finite state automaton takes this approach into account. But an expression for $\mathsf{Cl}_{\mathsf{Ab}}(\gamma(\varphi^{-1}(x)))$ can be obtained in a more efficient way as will be explained below.

The set $\gamma(\varphi^{-1}(x))$ is a *semilinear* subset of $\mathbb{Z}^n$, that is, a finite union of sets of the form $a + b_1\mathbb{N} + \cdots + b_p\mathbb{N}$, with $a, b_1, \ldots, b_p \in \mathbb{N}^n$. Such an expression for a semilinear set is said to be a *semilinear expression*. It was proved in [3] that the pro-Ab closure of $a + b_1\mathbb{N} + \cdots + b_p\mathbb{N}$, with $a, b_1, \ldots, b_p \in \mathbb{N}^n$ is the coset $a + b_1\mathbb{Z} + \cdots + b_r\mathbb{Z}$ of the subgroup of $\mathbb{Z}^n$ generated by the elements $b_1, \ldots, b_r$. Thus, to obtain the pro-Ab closure of a semilinear set given through a semilinear expression, what we have to do is to replace the $\mathbb{N}$'s by $\mathbb{Z}$'s, getting this way a so-called $\mathbb{Z}$-*semilinear expression* for a finite union of cosets of subgroups of $\mathbb{Z}^n$. A finite union of cosets of subgroups of $\mathbb{Z}^n$ is called a $\mathbb{Z}$-*semilinear set*. Note that the replacements done when computing the pro-Ab closure of a semilinear set correspond to substitute "submonoid generated by" by "subgroup generated by".

In [4] there are presented algorithms to compute semilinear expressions (respectively $\mathbb{Z}$-semilinear expressions) for $\gamma(\varphi^{-1}(x))$ (respectively $\mathsf{Cl}_{\mathsf{Ab}}(\gamma(\varphi^{-1}(x)))$) without the need of computing $\varphi^{-1}(x)$. Both consist on an adaptation of the state elimination algorithm which is possibly the most commonly used algorithm to compute a regular expression for the language recognized by a finite automaton. Recall that the state elimination algorithm basically consists in, at each state removal (which implies the elimination of the adjacent edges too), replace the labels of the edges remaining by regular expressions so that the generalized graph obtained recognizes the same language than the original automaton. (A generalized graph is similar to an automaton: the labels of its edges may be regular expressions instead of letters; the notion of recognition is obvious.) In the algorithms referred above to compute $\gamma(\varphi^{-1}(x))$ and $\mathsf{Cl}_{\mathsf{Ab}}(\gamma(\varphi^{-1}(x)))$ what is basically done is, at each state removal, instead of replacing the labels of the edges by regular expressions, one replaces them by expressions for its commutative images, in the first case, or by expressions for the pro-Ab closures of their commutative images, in the second case. The computation of these expressions for the pro-Ab closures involves the computation of the Hermite Normal Form of several matrices (depending on the adjacencies of the vertex to be removed), as a way to compute basis for the subgroups involved. In the final, as the computed subgroups are given through matrices in Hermite Normal Form they are, in particular, given by no more than n generators. Notice that this can not be the case for submonoids of $\mathbb{Z}^n$, since we may need much more that n elements to generate a submonoid of $\mathbb{Z}^n$, and this fact ultimately turns the computation of $\gamma(\varphi^{-1}(x))$ slower than the direct computation of $\mathsf{Cl}_{\mathsf{Ab}}(\gamma(\varphi^{-1}(x)))$, due to the memory required.

The paper [4] contains some details on the complexity of the algorithm presented to compute a $\mathbb{Z}$-*semilinear expression* for $\mathsf{Cl}_{\mathsf{Ab}}(\gamma(\varphi^{-1}(x)))$. It is exponential, but can be successfully used in practice to compute the abelian kernel of a finite monoid (see below for a definition). On the other hand, the problem of computing the abelian kernel of a finite monoid is polynomial: a polynomial time algorithm is given in [9], although that algorithm is presently only of theoretical interest, since it does not produce results in reasonable time.

4 Relative kernels of a finite monoid

Let S and T be monoids. A *relational morphism of monoids* $\tau : S \multimap T$ is a function from S into $\mathcal{P}(T)$, the power set of T, such that:

(a) For all $s \in S$, $\tau(s) \neq \emptyset$;

(b) For all $s_1, s_2 \in S$, $\tau(s_1)\tau(s_2) \subseteq \tau(s_1 s_2)$;

(c) $1 \in \tau(1)$.

A relational morphism $\tau : S \multimap T$ is, in particular, a relation in $S \times T$, and therefore we have a natural way to compose relational morphisms. As examples of relational morphism we have homomorphisms, seen as relations, and inverses of onto homomorphisms.

Given a pseudovariety H of groups, the H-*kernel of a finite monoid* S is the submonoid $\mathsf{K}_{\mathsf{H}}(S) = \bigcap \tau^{-1}(1)$, with the intersection being taken over all groups $G \in \mathsf{H}$ and all relational morphisms of monoids $\tau : S \multimap G$. When H is Ab, we use the terminology *abelian kernel*.

Let now M, φ and γ be as in Section 3. We adopt the usual notation for the neutral element of an abelian group: $(0, \ldots, 0) \in \mathbb{Z}^n$ is denoted by 0. The following holds [3, Proposition 5.3]:

Proposition 4.1 *Let $x \in M$. Then $x \in \mathsf{K}_{\mathsf{Ab}}(M)$ if and only if $0 \in \mathsf{Cl}_{\mathsf{Ab}}(\gamma(\varphi^{-1}(x)))$.*

4.1 The case of an infinite supernatural number

The following, analogous to Proposition 4.1, was proved by Steinberg [17, Proposition 6.1]:

Proposition 4.2 *Let π be an infinite supernatural number, H_π the associated pseudovariety of abelian groups and let $x \in M$. Then $x \in \mathsf{K}_{\mathsf{H}_\pi}(M)$ if and only if $0 \in \mathsf{Cl}_{\mathsf{H}_\pi}(\gamma(\varphi^{-1}(x)))$.*

The following lemma is crucial to compute $\mathsf{Cl}_{\mathsf{H}_\pi}(\gamma(\varphi^{-1}(x)))$ in the way we propose it next.

Lemma 4.3 *Let* H *and* H' *be pseudovarieties of groups such that* $\mathsf{H} \subseteq \mathsf{H}'$ *and let* $X \subseteq \mathbb{Z}^n$. *Then* $\mathsf{Cl}_{\mathsf{H}}(X) = \mathsf{Cl}_{\mathsf{H}}(\mathsf{Cl}_{\mathsf{H}'}(X))$.

PROOF. Since $\mathsf{H} \subseteq \mathsf{H}'$, we have $\mathsf{Cl}_{\mathsf{H}'}(X) \subseteq \mathsf{Cl}_{\mathsf{H}}(X)$. Thus $X \subseteq \mathsf{Cl}_{\mathsf{H}'}(X) \subseteq \mathsf{Cl}_{\mathsf{H}}(X)$. But then $\mathsf{Cl}_{\mathsf{H}}(X) \subseteq \mathsf{Cl}_{\mathsf{H}}(\mathsf{Cl}_{\mathsf{H}'}(X)) \subseteq \mathsf{Cl}_{\mathsf{H}}(\mathsf{Cl}_{\mathsf{H}}(X)) = \mathsf{Cl}_{\mathsf{H}}(X)$. □

As $\mathsf{H}_\pi \subseteq \mathsf{Ab}$ we immediately get the following result.

Corollary 4.4 $\mathsf{Cl}_{\mathsf{H}_\pi}(\gamma(\varphi^{-1}(x))) = \mathsf{Cl}_{\mathsf{H}_\pi}(\mathsf{Cl}_{\mathsf{Ab}}(\gamma(\varphi^{-1}(x))))$.

As we have already observed, a $\mathbb{Z}$-semilinear expression for $\mathsf{Cl}_{\mathsf{Ab}}(\gamma(\varphi^{-1}(x)))$ can be effectively computed. Having $\mathsf{Cl}_{\mathsf{Ab}}(\gamma(\varphi^{-1}(x)))$ expressed as a finite union of cosets of subgroups of $\mathbb{Z}^n$ one can use Corollary 4.4 to get $\mathsf{Cl}_{\mathsf{H}_\pi}(\gamma(\varphi^{-1}(x)))$ with very little extra computational work: replace each of the subgroups appearing in the expression for $\mathsf{Cl}_{\mathsf{Ab}}(\gamma(\varphi^{-1}(x)))$ by its pro-H_π closure proceeding as described in Section 2. Using the fact that the closure of the union is the union of the closures and using also the continuity of the group operation we conclude that what we obtain this way is precisely $\mathsf{Cl}_{\mathsf{H}_\pi}(\gamma(\varphi^{-1}(x)))$. Our problem of testing whether $0 \in \mathsf{Cl}_{\mathsf{H}_\pi}(\gamma(\varphi^{-1}(x)))$ is then reduced to test whether a diophantine system of linear equations has some solution, as also happened in the case of the abelian kernel. This can easily be done using GAP.

We observe that by following the proofs in [4] (see also Section 3) one could see that exactly the same way the state elimination algorithm is adapted to compute $\mathsf{Cl}_{\mathsf{Ab}}(\gamma(\varphi^{-1}(x)))$ it could be adapted to compute $\mathsf{Cl}_{\mathsf{H}_\pi}(\gamma(\varphi^{-1}(x)))$ directly. What had to be done was to consider expressions for the pro-H_π closures of semilinear sets instead of expressions for their pro-Ab closures. In practice, what we had to do was to replace the computations of Hermite Normal Forms by computations of Smith Normal Forms followed by computations of some greatest common divisors. As computing the Smith Normal Form of a matrix is slower than computing its Hermite Normal Form we believe that strategy proposed above, which makes use of the existing software, is not less efficient than a strategy that involves a more direct computation of $\mathsf{Cl}_{\mathsf{H}_\pi}(\gamma(\varphi^{-1}(x)))$. Notice that if we start by computing $\mathsf{Cl}_{\mathsf{Ab}}(\gamma(\varphi^{-1}(x)))$, the computation of Smith Normal Forms, as well as the computation of greatest common divisors, to obtain $\mathsf{Cl}_{\mathsf{H}_\pi}(\gamma(\varphi^{-1}(x)))$, needs to be performed only for the subgroups whose cosets appear in $\mathsf{Cl}_{\mathsf{Ab}}(\gamma(\varphi^{-1}(x)))$. In a direct computation of $\mathsf{Cl}_{\mathsf{H}_\pi}(\gamma(\varphi^{-1}(x)))$ we would need to compute Smith Normal Forms (rather than Hermite Normal Forms) for all subgroups whose cosets appear in intermediary expressions and this would turn the computation of $\mathsf{Cl}_{\mathsf{H}_\pi}(\gamma(\varphi^{-1}(x)))$ slower.

4.2 The case of a natural number

In this section, k is a natural number. We consider the projection $c_k : \mathbb{Z}^n \to (\mathbb{Z}/k\mathbb{Z})^n$ (defined by: $c_k(r_1, \ldots, r_n) = (r_1 \mod k, \ldots, r_n \mod k)$) and the homomorphism $\gamma_k = c_k \circ \gamma : A^* \to$

$(\mathbb{Z}/k\mathbb{Z})^n$. Note that for a word $w \in A^*$, the i^{th} component of $\gamma_k(w)$ is the number of occurrences modulo k of the i^{th} letter of A in w. Again, an analogous to Proposition 4.1 is obtained:

Proposition 4.5 *Let H_k be the pseudovariety of abelian groups associated to the natural number k, and let $x \in M$. Then $x \in \mathsf{K}_{\mathsf{H}_k}(M)$ if and only if $0 \in \gamma_k(\varphi^{-1}(x))$.*

PROOF. Notice that $(\mathbb{Z}/k\mathbb{Z})^n \in \mathsf{H}_k$ and the relation $\tau = \gamma_k \circ \varphi^{-1} : M \multimap (\mathbb{Z}/k\mathbb{Z})^n$ is a relational morphism. If $x \in \mathsf{K}_{\mathsf{H}_k}(M)$, we have that $x \in \tau^{-1}(0)$, thus $0 \in \gamma_k(\varphi^{-1}(x))$.

For the converse, let us consider a relational morphism $\mu : M \multimap G$, with $G \in \mathsf{H}_k$. Using the fact that $(\mathbb{Z}/k\mathbb{Z})^n$ is free relatively to H_k, it suffices to follow the proof of [3, Proposition 5.3] to see that there exists a homomorphism $\psi : (\mathbb{Z}/k\mathbb{Z})^n \to G$ such that $\mu = \psi \circ \gamma_k \circ \varphi^{-1}$. The conclusion follows then easily. □

Given a word $w \in A^*$ and a letter $a \in A$, we denote by $|w|_a$ the number of occurrences of the letter a in w. As an immediate consequence of the preceding proposition we have the following extremely simple characterization of the $\mathsf{K}_{\mathsf{H}_k}$-kernel of a finite monoid. We say that $w \in A^*$ *represents* $x \in M$ if $\varphi(w) = x$.

Corollary 4.6 *An element $x \in M$ is such that $x \in \mathsf{K}_{\mathsf{H}_k}(M)$ if and only if x can be represented by a word $w \in A^*$ such that $|w|_a \equiv 0 \mod k$, for any letter $a \in A$.*

From the fact that a submonoid of a finite group is in fact a subgroup we get the following:

Lemma 4.7 *Let N be a submonoid of $\mathbb{Z}^n$ and $\langle N \rangle$ the subgroup of $\mathbb{Z}^n$ generated by N. Then $c_k(N) = c_k(\langle N \rangle)$.*

Now, using the fact that $\gamma(\varphi^{-1}(x))$ is a semilinear set and that the pro-Ab closure of a semilinear set is obtained replacing "submonoid generated by" by "subgroup generated by" we get:

Corollary 4.8 *Let $x \in M$. Then $c_k(\gamma(\varphi^{-1}(x))) = c_k(\mathsf{Cl}_{\mathsf{Ab}}(\gamma(\varphi^{-1}(x))))$.*

As a consequence we have that we can use an expression of $\mathsf{Cl}_{\mathsf{Ab}}(\gamma(\varphi^{-1}(x)))$ as a finite union of cosets of subgroups of $\mathbb{Z}^n$ to compute $\gamma_k(\varphi^{-1}(x))$. Recall that our problem is to test whether 0 belongs to $\gamma_k(\varphi^{-1}(x))$. This can now be reduced to test whether a system of linear equations has some solution in $(\mathbb{Z}/k\mathbb{Z})^n$, which is not difficult to do using GAP.

One could expect that, due to its simplicity, the direct usage of Corollary 4.6 to test whether $x \in \mathsf{K}_{\mathsf{H}_k}(M)$ would be very fast. But we must note that it would require to compute an expression for $\gamma(\varphi^{-1}(x))$ which, as referred in Section 4, is less efficient than computing $\mathsf{Cl}_{\mathsf{Ab}}(\gamma(\varphi^{-1}(x)))$. So, again, we believe that the better strategy is to start computing $\mathsf{Cl}_{\mathsf{Ab}}(\gamma(\varphi^{-1}(x)))$, which corresponds to make use of the existing software.

References

[1] C.J. Ash, *Inevitable graphs: a proof of the type II conjecture and some related decision procedures*, Int. J. Algebra Comput. **1** (1991) 127–146.

[2] H. Cohen, "A Course in Computational Algebraic Number Theory", GTM, Springer Verlag, 1993.

[3] M. Delgado, *Abelian pointlikes of a monoid*, Semigroup Forum **56** (1998) 339–361.

[4] M. Delgado, *Commutative images of rational languages and the abelian kernel of a monoid*, Theor. Inform. Appl. **35** (2001) 419–435.

[5] M. Delgado and V.H. Fernandes, *Abelian kernels of some monoids of injective partial transformations and an application*, Semigroup Forum **61** (2000) 435–452.

[6] M. Delgado and V.H. Fernandes, *Solvable monoids with commuting idempotents,* Int. J. Algebra Comput. To appear.

[7] M. Delgado and V.H. Fernandes, *Abelian kernels, solvable monoids and the abelian kernel length of a finite monoid*, in "Proceedings of the Workshop on Semigroups and Languages" (Lisbon 2002), I. Araújo, M. Branco, V. Fernandes, G. Gomes (eds.) World Scientific, (2004), 68-85.

[8] M. Delgado and V.H. Fernandes, *Abelian kernels of monoids of order-preserving maps and of some of its extensions*, Semigroup Forum. **68** (2004) 435–456.

[9] M. Delgado and P.-C. Héam, *A polynomial time algorithm to compute the Abelian kernel of a finite monoid*, Semigroup Forum, **67** (2003) 97–110.

[10] M. Delgado, S. Linton and J. Morais, *Automata: a* GAP *[18] package on finite automata.* (http://www.gap-system.org/Packages/automata.html).

[11] M. Delgado and J. Morais, *Approximation to the smallest regular expression for a given regular language*, in "Implementation and Applicationof Automata",M Domaratzki, A. Okhotin, K. Salomaa and S. Yu, (eds), volume 3317 LNCS. Springer-Verlag, Heidelberg, January 2005.

[12] K. Henckell, S. Margolis, J.-E. Pin and J. Rhodes, *Ash's type II theorem, profinite topology and Malcev products. Part I*, Int. J. Algebra Comput. **1** (1991) 411–436.

[13] J.-E. Pin and C. Reutenauer, *A conjecture on the Hall topology for the free group*, Bull. London Math. Soc. **23** (1991) 356–362.

[14] L. Ribes and P.A. Zalesskiĭ, *On the profinite topology on a free group,* Bull. London Math. Soc. **25** (1993) 37–43.

[15] L. Ribes and P.A. Zalesskiĭ, *The pro-p topology of a free group and algorithmic problems in semigroups* Int. J. Algebra Comput. **4** (1994) 359–374.

[16] C. C. Sims, "Computation with Finitely Presented Groups", Cambridge University Press, 1994.

[17] B. Steinberg, *Monoid kernels and profinite topologies on the free Abelian group*, Bull. Austral. Math. Soc. **60** (1999) 391–402.

[18] The GAP Group. *GAP – Groups, Algorithms, and Programming, Version 4.4*, 2004. (`http://www.gap-system.org`).

Edite Cordeiro
Instituto Politécnico de Bragança
Escola Superior de Tecnologia e Gestão
Campus de Santa Apolónia
5301-857 Bragança
Portugal
e-mail: emc@ipb.pt

Manuel Delgado
Centro de Matemática
Universidade do Porto
Rua do Campo Alegre, 687
4169-007 Porto
Portugal
e-mail: mdelgado@fc.up.pt