MATHEMATICS OF COMPUTATION
Volume 74, Number 252, Pages 2017–2026
S 0025-5718(05)01740-0
Article electronically published on March 8, 2005

COMPUTING THE STRUCTURE OF A FINITE ABELIAN GROUP

JOHANNES BUCHMANN AND ARTHUR SCHMIDT

Abstract. We present an algorithm that computes the structure of a finite abelian group G from a generating system M. The algorithm executes $\mathrm{O}(|M|\sqrt{|G|})$ group operations and stores $\mathrm{O}(\sqrt{|G|})$ group elements.

1. Introduction

Let G be a finite abelian group. Then G can be written as a direct product

$$G = \langle G_1 \rangle \times \cdots \times \langle G_k \rangle, \tag{1.1}$$

where k is a positive integer, G_i is a cyclic subgroup of G, $1 \le i \le k$, and if n_i is the order of G_i, $1 \le i \le k$, then n_i divides n_{i+1} for $1 \le i < k$. The integers n_i are uniquely determined by G. They are called the *invariants* of G. Let n be the order of G. Let

$$\varphi : G \to \{1, \ldots, n^c\} \tag{1.2}$$

for some positive integer c. Suppose that we can compute $\varphi(a \cdot b)$ from $\varphi(a)$ and $\varphi(b)$ and $\varphi(a^{-1})$ from $\varphi(a)$ for all $a, b \in G$. Those are the *group operations*.

The *group structure problem* is the following: Given a *generating system* for G, that is, a sequence $(g_1, \ldots, g_l)$ of group elements such that $G = \{\prod_{i=1}^{l} g_i^{e_i} : e_i \in \mathbb{Z}, 1 \le i \le l\}$, compute the invariants $n_1, \ldots, n_k$ of G and group elements h_i, $i \le i \le k$, such that $|\langle h_i \rangle| = n_i$, $1 \le i \le k$ and the cyclic subgroups $G_i = \langle h_i \rangle$ generated by the h_i satisfy (1.1).

The fastest algorithm [BJT97] for the group structure problem known so far executes $|M|2^{k/2}|G|^{1/2+o(1)}$ group operations and stores $|G|^{1/2+o(1)}$ group elements, where $o(1)$ is a function that goes to zero as $|G|$ goes to infinity . In this paper, we present a new algorithm that allows us to prove the following theorem.

Theorem 1.1. *Computing the structure of the finite abelian group G from the generating system M requires storing $\mathrm{O}(\sqrt{|G|})$ pairs $(g, \vec{q}) \in G \times \{0, \ldots, \lfloor\sqrt{|G|}\rfloor\}^{|M|}$, $\mathrm{O}(|M|\sqrt{|G|})$ multiplications and inversions in G, $\mathrm{O}(|M|\sqrt{|G|})$ table lookups, and $(|M| \log |G|)^{\mathrm{O}(1)}$ bit operations.*

The new algorithm is no longer exponential in the number of generators. The algorithm is based on an idea of Terr [Ter00] for computing the order of an element of G.

Received by the editor April 23, 2003 and, in revised form, August 2, 2004.
2000 *Mathematics Subject Classification.* Primary 11Y16; Secondary 20C40, 20K02.

The paper is organized as follows. In section 2 we describe Terr's algorithm for computing the order of a group element $g \in G$. In section 3 we present the new algorithm for computing the structure of G.

2. Computing the order of an element

In this section we present an algorithm for computing the order $\operatorname{order}(g)$ of an element $g \in G$. This algorithm is a special case of an algorithm by Terr [Ter00] and is based on the following statement.

Lemma 2.1. *Let $g \in G$. Then there is $e \in \mathbb{N}$ and $f \in \{0, \dots, e-1\}$ with $g^{e(e+1)/2} = g^f$. If e is chosen minimal with this property, then $e(e-1)/2 < \operatorname{order}(g) \le e(e+1)/2$, f is uniquely determined, and $\operatorname{order}(g) = e(e+1)/2 - f$.*

Proof. Let $e \in \mathbb{N}$ such that $e(e-1)/2 < \operatorname{order}(g) \le e(e+1)/2$. Since $e(e-1)/2+e = e(e+1)/2$, such an e exists. Let $f = e(e+1)/2 - \operatorname{order}(g)$. Then $f \in \{0, \dots, e-1\}$. Also, since $g^{e(e+1)/2-f} = g^{\operatorname{order}(g)} = 1$, it follows that $g^{e(e+1)/2} = g^f$. This proves the existence of e and f.

We prove the minimality of e. Let $e' \in \mathbb{N}$, $f' \in \{0, \dots, e'-1\}$ such that $g^{e'(e'+1)/2-f'} = 1$. Then $e'(e'+1)/2 \ge e'(e'+1)/2 - f' \ge \operatorname{order}(g) = e(e+1)/2 - f > e(e-1)/2$. Since e and e' are integers, this implies $e'(e'-1)/2 \ge e(e-1)/2$. Hence, $e' \ge e$. □

For $e = 1, 2, \dots$ Terr's algorithm computes the set

$$\mathsf{babySet} = \{(g^f, f) : 0 \le f < e\} \tag{2.1}$$

and checks whether there exists a pair of the form $(g^{e(e+1)/2}, f)$ in $\mathsf{babySet}$ for some f. By Lemma 2.1 this will eventually happen. If this happens for the first time, then we have $\operatorname{order}(g) = e(e+1)/2 - f$. In the algorithm we use

$$\mathsf{babyElement} = g^e, \quad \mathsf{giantElement} = g^{e(e+1)/2}.$$

Here is the algorithm.

Algorithm 1. `order` (g)

Input: A group element g.
Output: The order n of g.

$\mathsf{babySet} \leftarrow \{(1, 0)\}$.
$e \leftarrow 1$
$\mathsf{babyElement} \leftarrow g$
$\mathsf{giantElement} \leftarrow g$
loop
  if $\mathsf{babySet}$ contains a pair $(\mathsf{giantElement}, f)$ **then** return $n = e(e+1)/2 - f$
  insert $(\mathsf{babyElement}, e)$ in $\mathsf{babySet}$
  $\mathsf{babyElement} \leftarrow g \cdot \mathsf{babyElement}$
  $\mathsf{giantElement} \leftarrow \mathsf{giantElement} \cdot \mathsf{babyElement}$
  $e \leftarrow e + 1$

We analyze Terr's algorithm.

Theorem 2.2. *Let $g \in G$ and let $n = \operatorname{order}(g)$. Given g, algorithm* `order`(g) *terminates and returns n. Algorithm* `order`(g) *executes at most $\sqrt{2n} + 1/2$ iterations, $2\sqrt{2n} - 1$ multiplications in G and $\sqrt{2n} + 1/2$ table lookups. Also, algorithm* `order`(g) *stores at most $\sqrt{2n} + 1/2$ elements of G.*

Proof. It follows from Lemma 2.1 that `order` terminates and upon termination we have $e(e-1)/2 < n \le e(e+1)/2$. Since e and n are integers we have $(e - 1/2)^2 = e(e-1) + 1/4 < 2n$, which implies $e < \sqrt{2n} + 1/2$. In the first $e-1$ iterations of the while loop, 2 multiplications are executed. In the last iteration no multiplication is performed. Also, table babySet is accessed twice in each iteration, once to test whether $(\mathsf{giantElement}, f) \in \mathsf{babySet}$, and once to store the pair $(\mathsf{babyElement}, e)$ in babySet. Since the number of iterations is at most $\sqrt{2n} + 1/2$, this implies the assertion. □

3. Computing the structure

Let

$$M = (g_1, \ldots, g_l)$$

be a generating system for G. For $\vec{q} = (q_1, \ldots, q_l) \in \mathbb{Z}^l$ we write

$$M^{\vec{q}} = \prod_{j=1}^{l} g_j^{q_j}.$$

A *relation* for M is a vector $\vec{q} \in \mathbb{Z}^l$ such that $M^{\vec{q}} = 1$. The set $L(M)$ of all relations for M is a lattice in $\mathbb{Z}^l$. Since that lattice is the kernel of the surjective homomorphism

$$\mathbb{Z}^l \longrightarrow G, \quad \vec{q} \mapsto M^{\vec{q}}, \tag{3.1}$$

its dimension is l. Also, if $U = (\vec{u}_1, \ldots, \vec{u}_m) \in \mathbb{Z}^{l,m}$, then we write

$$M^U = (M^{\vec{u}_1}, \ldots, M^{\vec{u}_m}).$$

Our algorithm is based on the following lemma.

Lemma 3.1. *Let $B = (\vec{b}_1, \ldots, \vec{b}_l) \in \mathbb{Z}^{(l,l)}$ be a basis of the relation lattice $L(M)$. Let $D \in \mathbb{Z}^{(l,l)}$ be the Smith normal form of B. Let $D = \operatorname{diag}(n_1, \ldots, n_k, 1, \ldots, 1)$ with $s_k > 1$. Let $U'D = BV$ with $U', V \in \mathrm{GL}(l, \mathbb{Z})$. Let $U \in \mathbb{Z}^{(l,l)}$ with $U \equiv U' \pmod{|G|}$. Let $M^U = (h_1, \ldots, h_k, h_{k+1}, \ldots, h_l)$. Then the following are true.*

1. *The order of G is $|\det B|$.*
2. *The invariants of G are $n_1, \ldots, n_k$.*
3. *The order of h_i is n_i, $1 \le i \le k$, and*

$$G = \langle h_1 \rangle \times \cdots \times \langle h_k \rangle. \tag{3.2}$$

Proof. 1. The determinant of B is the index of the kernel of the map (3.1) in $\mathbb{Z}^l$. That index is the order of G.

2. and 3. We claim that M^U is a generating system for G. Clearly, we have $(M^U)^{\vec{v}} \in G$ for any $\vec{v} \in \mathbb{Z}^l$. Conversely, let $g \in G$. Then there is $\vec{v} \in \mathbb{Z}^l$ with $g = M^{\vec{v}}$. Since $\gcd(\det U, |G|) = 1$, it follows that there is $\tilde{U} \in \mathbb{Z}^{(l,l)}$ such that $U\tilde{U} \equiv I_l \pmod{|G|}$ where I_l is the $l \times l$-identity matrix. Set $\tilde{\vec{v}} = \tilde{U}\vec{v}$. Then

$$(M^U)^{\tilde{\vec{v}}} = (M^U)^{\tilde{U}\vec{v}} = M^{U\tilde{U}\vec{v}} = M^{\vec{v}} = g.$$

This proves our claim.

Next, we show that the columns of D form a basis for the relation lattice of M^U. Since $(M^U)^D = M^{UD} = M^{U'D} = M^{BV}$, it follows that the columns of D are relations for M^U. Let $\vec{v}$ be a relation for M^U. Then $1 = (M^U)^{\vec{v}} M^{U'\vec{v}}$. It follows that $U'\vec{v}$ is a relation for M. Since BV is a basis for $L(M)$, there is $\vec{x} \in \mathbb{Z}^l$ with $U'\vec{v} = BV\vec{x} = U'D\vec{x}$. Hence, $\vec{v} = D\vec{x}$. This proves that D is a basis of $L(M^U)$. It follows that the ith diagonal element d_i of D is the order of h_i, $1 \le i \le l$. In particular, we have $h_{k+1} = \ldots = h_l = 1$. This implies that $(h_1, \ldots, h_k)$ is a generating system for G.

Since D is a basis of $L(M^U)$, it follows that if $(h_1, \ldots, h_k)^{\vec{e}} = 1$ for some $\vec{e} \in \mathbb{Z}^k$, then $e_i \equiv 0 \pmod{n_i}$ where e_i is the ith entry in $\vec{e}$, $1 \le i \le k$. This proves (3.2). Since D is the Smith normal form of B, it follows that $n_1, \ldots, n_k$ are the invariants of G. □

By this lemma, the structure of G can be computed as follows. We determine a basis B of the relation lattice $L(M)$. We use standard techniques (e.g. [HM91]) to compute the Smith normal form D and a matrix $U \in \mathbb{Z}^{(l,l)}$ with the properties from Lemma 3.1. Then the invariants and the representation of G as a product of cyclic groups whose orders are the invariants can be computed as described in the lemma.

We describe the computation of the basis $B = (\vec{b}_1, \ldots, \vec{b}_l)$ of the relation lattice $L(M)$. We write

$$\vec{b}_j = (b_{1,j}, \ldots, b_{l,j}).$$

The matrix B will be in Hermite normal form, that is, $b_{i,j} = 0$ for $0 \le j < i \le l$ and $0 < b_{i,j} < b_{j,j}$, $1 \le i < j \le l$.

Let $j \in \mathbb{Z}$, $1 \le j \le k$ and suppose that we have computed the basis vectors $\vec{b}_1, \ldots, \vec{b}_{j-1}$. We describe the computation of $\vec{b}_j$. The idea is as follows. Let

$$\vec{e}_i = (\underbrace{0, \ldots, 0}_{i-1}, 1, \underbrace{0, \ldots, 0}_{l-i}) \in \mathbb{Z}^l, \quad 1 \le i \le l.$$

The subgroup H generated by the $g_1, \ldots, g_{j-1}$ is

$$H = \{\prod_{i=1}^{j-1} g_i^{x_i} : 0 \le x_i < b_{i,i}, 1 \le i < j\}. \tag{3.3}$$

Note that H depends on j. But for simplicity, we omit the index j. The entry $b_{j,j}$ is the order of the coset $g_j H$ in the factor group G/H. So we can use the order algorithm from the previous section to calculate that entry. We have to look for the smallest e such that

$$g_j^{e(e+1)/2} = g_j^f h \tag{3.4}$$

for some $f \in \{0, \ldots, e-1\}$ and some $h \in H$. As in algorithm `order`, we could store the values on the right-hand side and try to find an e that satisfies (3.4). However, H can be as large as the whole group G. This is too large to obtain the complexity that we want. Therefore, we split H into two parts. We use a decomposition

$$\{1, \ldots, j-1\} = I_1 \cup \{m\} \cup I_2, \tag{3.5}$$

where the three sets on the right-hand side are pairwise disjoint. Let

$$H_1 = \{(M^{-\vec{v}}, \vec{v}) : \vec{v} = \sum_{i \in I_1} x_i e_i, 0 \le x_i < b_{i,i}, i \in I_1\} \tag{3.6}$$

and

$$H_2 = \{(M^{\vec{v}}, \vec{v}) : \vec{v} = \sum_{i \in I_2} x_i e_i, 0 \le x_i < b_{i,i}, i \in I_2\}. \tag{3.7}$$

The decomposition in (3.5) is chosen such that

$$|H_i| \le \sqrt{|H|}, \quad i = 1, 2. \tag{3.8}$$

We set

$$s = \left\lceil \sqrt{|H|}/|H_1| \right\rceil \tag{3.9}$$

and

$$t = \left\lceil \sqrt{|H|}/|H_2| \right\rceil. \tag{3.10}$$

Now we have the following result.

Lemma 3.2. *Any $h \in H$ can be written as $h = h_1^{-1} g_m^{qs+r} h_2$, where h_i is the first entry of a pair in H_i, $i = 1, 2$ and we have $0 \le q < t$ and $0 \le r < s$.*

Proof. By (3.3) and (3.5) we can write

$$h = h_1^{-1} g_m^n h_2,$$

where h_i is the first entry of a pair in H_i, $i = 1, 2$ and $n \in \{0, \ldots, b_{m,m} - 1\}$. Write $n = qs + r$ with $0 \le r < s$. Then $qs < b_{m,m}$. Hence $q < b_{m,m}/s \le b_{m,m}|H_1|/\sqrt{|H|} = |H|/(|H_2|\sqrt{|H|}) \le t$. □

Now we modify (3.4). To find $\vec{b}_j$ we look for the smallest e such that

$$g_j^{e(e+1)/2} h_2 g_m^{qs} = g_j^f h_1 g_m^{-r}, \tag{3.11}$$

where $(h_i, \vec{v}_i) \in H_i$ for some $\vec{v}_i$, $i = 1, 2$, $0 \le r < s$, $0 \le q < t$, and $0 \le f < e$. Then

$$\vec{b}_j = \vec{v}_1 + \vec{v}_2 + (qs + r)\vec{e}_m + (e(e+1)/2 - f)\vec{e}_j. \tag{3.12}$$

To look for a match of the form (3.11) we use two sets. The first one is

$$\mathsf{babySet} = \{(g_j^f h, \vec{v} - f\vec{e}_j) : (h, \vec{v}) \in \mathsf{auxiliaryBabySet}, 0 \le f < e\},$$

where

$$\mathsf{auxiliaryBabySet} = \{(h_1 g_m^{-r}, \vec{v} + r\vec{e}_m) : (h_1, \vec{v}) \in H_1, 0 \le r < s\}.$$

In the set babySet we store the elements from the right side of (3.11). The second set is

$$\mathsf{giantSet} = \{(h_2 g_m^{qs}, \vec{v} + qs\vec{e}_m) : (h_2, \vec{v}) \in H_2, 0 \le q < t\}.$$

As in the order algorithm we use

$$\mathsf{babyElement} = g_j^e, \quad \mathsf{giantElement} = g_j^{e(e+1)/2}.$$

In iteration e we multiply giantElement with each element of giantSet and check whether the product is in babySet. If so, we have the match that we have looked for and can compute $\vec{b}_j$. If there is no such match, we increment e, update babySet, babyElement and giantElement and repeat the procedure. If $\vec{b}_j$ has been determined and $j = l$, then the algorithm terminates. If $\vec{b}_j$ has been determined and $j < l$, then a new decomposition (3.5) is determined and the sets I_1, I_2, H_1, H_2, auxiliaryBabySet, and giantSet are updated. If $b_{j,j} = 1$, then the decomposition (3.5) and the sets I_1, I_2, H_1, H_2, auxiliaryBabySet remain unchanged. The treatment of the other cases can be seen in the algorithm.

ALGORITHM 2. `HNFRelationBasis` (M)

Input: A system $M = (g_1, \dots, g_l)$ of group elements
Output: The HNF-Basis $B = (\vec{b}_1, \dots, \vec{b}_l)$ of $L(M)$

$H_i \leftarrow \{\big(1, (0, \dots, 0)\big)\}$, $i = 1, 2$
$I_i \leftarrow \emptyset$, $i = 1, 2$
$s \leftarrow 0$, $t \leftarrow 0$, $m \leftarrow 0$
auxiliaryBabySet $\leftarrow H_1$; giantSet $\leftarrow H_2$
for $j = 1, \dots, l$ **do**
  $e \leftarrow 1$
  babySet $\leftarrow$ auxiliaryBabySet, babyElement $\leftarrow g_j$, giantElement $\leftarrow g_j$
  loop
    for all $(g, \vec{v}) \in$ giantSet **do**
      if babySet contains a pair $(g \cdot$ giantElement$, \vec{w})$ **then**
        $\vec{b}_j \leftarrow \vec{v} + \vec{w} + (e(e+1)/2)\vec{e}_j$
        break
    babySet $\leftarrow$ babySet $\cup \{(g \cdot$ babyElement$, \vec{v} - e\vec{e}_j) : \ (g, \vec{v}) \in$ auxiliaryBabySet$\}$
    $e \leftarrow e + 1$; babyElement $\leftarrow$ babyElement $\cdot\, g_j$;
    giantElement $\leftarrow$ giantElement $\cdot$ babyElement
  if $j < l$ and $b_{j,j} > 1$ **then**
    if $b_{j,j} \prod_{i \in I_1} b_{i,i} \leq \sqrt{\prod_{i=1}^{j} b_{i,i}}$ **then**
      $H_1 \leftarrow H_1 \cup \{(g_j^{-x} g, \vec{v} + x\vec{e}_j) : (g, \vec{v}) \in H_1, 0 \leq x < b_{j,j}\}$
      $I_1 \leftarrow I_1 \cup \{j\}$
    else
      if $m > 0$ **then**
        $H_2 \leftarrow H_2 \cup \{(g_m^x g, \vec{v} + x\vec{e}_m) : (g, \vec{v}) \in H_2, 0 \leq x < b_{m,m}\}$
        $I_2 \leftarrow I_2 \cup \{m\}$
      $m \leftarrow j$
    $s \leftarrow \lceil \sqrt{\prod_{i=1}^{j} b_{i,i}} / \prod_{i \in I_1} b_{i,i} \rceil$
    $t \leftarrow \lceil \sqrt{\prod_{i=1}^{j} b_{i,i}} / \prod_{i \in I_2} b_{i,i} \rceil$
    auxiliaryBabySet $\leftarrow \{(h_1 g_m^{-r}, \vec{v} + r\vec{e}_m) : (h_1, \vec{v}) \in H_1, 0 \leq r < s\}$
    giantSet $\leftarrow \{(h_2 g_m^{qs}, \vec{v} + qs\vec{e}_m) : (h_2, \vec{v}) \in H_2, 0 \leq q < t\}$
return $(\vec{b}_1, \dots, \vec{b}_k)$

In the analysis of the structure algorithm we need the following lemma.

Lemma 3.3. (1) *Let $k \in \mathbb{N}$ and $a_1, \dots, a_k \in \mathbb{R}_{\geq 1}$. Then*
$\sum_{j=1}^{k} a_j \leq \prod_{j=1}^{k} a_j + (k-1)$.
(2) *Let $k \in \mathbb{N}$ and $a_1, \dots, a_k \in \mathbb{R}_{\geq 2}$. Then*
$\sum_{j=1}^{k} \prod_{i=1}^{j} \sqrt{a_j} \leq (2 + \sqrt{2}) \prod_{j=1}^{k} \sqrt{a_j}$.

Proof. 1. For any $x, y \in \mathbb{R}_{\geq 1}$ we have

$$x + y - xy - 1 = \underbrace{(x-1)}_{\geq 0}\underbrace{(1-y)}_{\leq 0} \leq 0.$$

Hence

$$x + y \le xy + 1. \tag{3.13}$$

Now, we prove the first statement of the lemma by induction.

For $k = 1$, the assertion is true.

Assume that the statement is true for $k - 1$. Then we have

$$\sum_{j=1}^{k} a_j = \sum_{j=1}^{k-1} a_j + a_k \le \underbrace{\prod_{j=1}^{k-1} a_j}_{\ge 1} + a_k + (k-2) \le \prod_{j=1}^{k} a_j + (k-1).$$

2. Let $B = \prod_{j=1}^{k} \sqrt{a_j}$. Then

$$\begin{aligned}\sum_{j=1}^{k} \left(\prod_{i=1}^{j} \sqrt{a_i} \right) &= B \sum_{j=1}^{k} \left(\prod_{i=j+1}^{k} \frac{1}{\sqrt{a_i}} \right) \le B \sum_{j=1}^{k} \left(\prod_{i=j+1}^{k} \frac{1}{\sqrt{2}} \right) \\ &= B \sum_{j=1}^{k} \left(\frac{1}{\sqrt{2}} \right)^{k-j} = B \sum_{j=0}^{k-1} \left(\frac{1}{\sqrt{2}} \right)^{j} = B \frac{1 - (1/\sqrt{2})^k}{1 - 1/\sqrt{2}} \\ &= B(2 + \sqrt{2} - \frac{2 + \sqrt{2}}{\sqrt{2}^k}) \le (2 + \sqrt{2}) \prod_{j=1}^{k} \sqrt{a_j}. \end{aligned}$$

□

We now present the main result of this paper. By $l(M)$ we denote the number of diagonal entries in the HNF-basis of $L(M)$ that are greater than 1.

Theorem 3.4. *Algorithm* `HNFRelationBasis` *computes the HNF-basis of the lattice of relations on M and executes*

- $l = |M|$ *inversions,*
- *at most* $(48 + 8l - 6l(M))\sqrt{|G|} + 2l(M) \log \sqrt{|G|}$ *multiplications in G,*
- *at most* $4(2 + \sqrt{2} + l - l(M))\sqrt{|G|}$ *table lookups.*

The algorithm uses

- *two tables of at most* $\sqrt{|G|}$,
- *two tables of at most* $2\sqrt{|G|}$,
- *one table of at most* $4\sqrt{|G|}$

pairs $(g, \vec{q}) \in G \times \{0, \ldots, \lfloor \sqrt{|G|} \rfloor\}^{|M|}$.

Proof. We first estimate the sizes of the sets H_1, H_2, babySet, auxiliaryBabySet, and giantSet. Then we estimate the number of group operations and table lookups.

Consider the computation of $\vec{b}_j$. Let $e(j)$ be the final value for e in the computation of $\vec{b}_j$. By Lemma 2.2 we have

$$e(j) < \sqrt{2b_{j,j}} + 1/2 \le 2\sqrt{b_{j,j}}.$$

First, `HNFRelationBasis` computes the order of the coset $g_j H$ in the factor group G/H and the vector $\vec{b}_j$. Then, it updates the sets H_1, H_2, auxiliaryBabySet, and giantSet for the next loop.

We analyze the first step. By (3.8) we have

$$|H_i| \le \sqrt{|H|} = \sqrt{\prod_{i=1}^{j} b_{i,i}} \le \sqrt{|G|}, \quad i = 1, 2. \tag{3.14}$$

It follows from (3.9) and (3.10) that

$$|\mathsf{auxiliaryBabySet}| = s|H_1| \le 2\sqrt{|G|} \text{ and } |\mathsf{giantSet}| = t|H_2| \le 2\sqrt{|G|}. \tag{3.15}$$

The set babySet is constructed from the set auxiliaryBabySet from the $(j-1)$th iteration. Therefore we have

$$\begin{aligned} |\mathsf{babySet}| &\le e(j)|\mathsf{auxiliaryBabySet}| \\ &\le 4\sqrt{b_{j,j}}\sqrt{\prod_{i=1}^{j-1} b_{j,j}} \le 4\sqrt{|G|}. \end{aligned}$$

We estimate the number of table lookups. We consider the cases $e(j) = 1$ and $e(j) > 1$. In the first case, we have $b_{j,j} = 1$. In the second case we have $b_{j,j} \ge 2$. By Lemma 3.3 we have at most

$$\begin{aligned} \sum_{j=1}^{l} e(j)|\mathsf{giantSet}| &\le \sum_{j=1}^{l} 2e(j)\sqrt{\prod_{i=1}^{j-1} b_{i,i}} \le \sum_{j=1}^{l} 4\prod_{i=1}^{j}\sqrt{b_{i,i}} \\ &\le 4\sum_{j=1,e(j)>1}^{l}\prod_{i=1}^{j}\sqrt{b_{i,i}} + 4\sum_{j=1,e(j)=1}^{l}\prod_{i=1}^{j}\sqrt{b_{i,i}} \\ &\le 4(2+\sqrt{2})\prod_{j=1,e(j)>1}^{l}\sqrt{b_{i,i}} + 4(l-l(M))\prod_{j=1}^{l}\sqrt{b_{i,i}} \\ &\le 4(2+\sqrt{2}+l-l(M))\sqrt{|G|} \end{aligned}$$

table lookups.

We estimate the number of multiplications. The number of multiplications necessary to multiply all first elements of giantSet by giantElement is

$$\begin{aligned} M_1 &\le \sum_{j=1}^{l} e(j)|\mathsf{giantSet}| \\ &\le 4(2+\sqrt{2}+l-l(M))\sqrt{|G|}. \end{aligned}$$

The number of multiplications to update babySet is

$$\begin{aligned} M_2 &\le \sum_{j=1}^{l} e(j)|\mathsf{auxiliaryBabySet}| \\ &\le 4(2+\sqrt{2}+l-l(M))\sqrt{|G|}. \end{aligned}$$

The number of multiplications necessary to update babyElement and giantElement is

$$\begin{aligned}
M_3 &\le \sum_{j=1}^{l}(2e(j)-2) \le 2\sum_{j=1}^{l} e(j) - 2l \\
&= 2\sum_{j=1,e(j)>1}^{l} e(j) + 2\sum_{j=1,e(j)=1}^{l} e(j) - 2l \\
&\le 2\sum_{j=1,e(j)>1}^{l} (\sqrt{2b_{j,j}} + 1/2) + 2(l - l(M)) - 2l \\
&\le 2\sqrt{2}\sum_{j=1,e(j)>1}^{l} \sqrt{b_{j,j}} + l(M) - 2l(M) \\
&\le 2\sqrt{2}\prod_{j=1,e(j)>1}^{l} \sqrt{b_{j,j}} + 2\sqrt{2}(l(M)-1) - l(M) \text{ (Lemma 3.3)} \\
&\le 2\sqrt{2|G|} + 2l(M).
\end{aligned}$$

Now we analyze the number of multiplications necessary to update H_1, H_2, auxiliaryBabySet, and giantSet. No multiplications are executed if $\vec{b}_{j,j} = 1$. In each loop with $\vec{b}_{j,j} > 1$ either $|H_1|$ multiplications are necessary to update H_1 or $|H_2|$ multiplications are necessary to update H_2. Next, |auxiliaryBabySet|+|giantSet| multiplications are used to update auxiliaryBabySet and babySet, and finally at most $2\lfloor \log \sqrt{|G|} \rfloor$ multiplications are performed to compute g_m^s during the computation of TG. By (3.14), (3.15) and Lemma 3.3, the total number of multiplications required for those updates is

$$\begin{aligned}
M_4 &\le \sum_{j=1}^{l}(5\sqrt{\prod_{i=1}^{j} b_{i,i}} + 2\lfloor \log \sqrt{|G|} \rfloor) \\
&\le 5(2+\sqrt{2})\sqrt{|G|} + 2l(M)\log\sqrt{|G|}.
\end{aligned}$$

So the number of multiplications is

$$\begin{aligned}
M &\le M_1 + M_2 + M_3 + M_4 \\
&\le (48 + 8l - 6l(M))\sqrt{|G|} + 2l(M)\log\sqrt{|G|}.
\end{aligned}$$

Finally the algorithm executes l inversion. □

Theorem 1.1 can be deduced from Theorem 3.4. The number of group operations and table lookups is estimated in Theorem 3.4. We estimate the number of bit operations. When the algorithm updates the sets auxiliaryBabySet, babySet, giantSet, H_1, and H_2 it executes $|M|\sqrt{|G|}(\log|G|)^{O(1)}$ bit operations. By [HM91] the modular computation of the Smith normal forms is possible in time $(|M|\log|G|)^{\mathrm{O}(1)}$ since that algorithm has polynomial running time, the entries of the HNF-basis are in $\{0,\ldots,|G|\}$, and the dimension of that matrix is $|M| \times |M|$.

References

[BJT97] J. Buchmann, M.J. Jacobson, Jr., and E. Teske, *On some computational problems in finite abelian groups*, Mathematics of Computation **66** (1997), 1663–1687. MR1432126 (98a:11185)

[HM91] J.L. Hafner and K.S. McCurley, *Asymptotically fast triangularization of matrices over rings*, SIAM Journal on Computing **20** (1991), 1068–1083. MR1135749 (93d:15021)

[Ter00] David C. Terr, *A modification of Shanks' baby-step giant-step algorithm*, Math. Comp. **69** (2000), no. 230, 767–773. MR1653994 (2000i:20039)

Technische Universität Darmstadt, Theoretische Informatik, Hochschulstr. 10, 64289 Darmstadt, Germany

E-mail address: buchmann@cdc.informatik.tu-darmstadt.de

Technische Universität Darmstadt, Theoretische Informatik, Hochschulstr. 10, 64289 Darmstadt, Germany

E-mail address: aschmidt@cdc.informatik.tu-darmstadt.de