Chapter 2

Linear algebra over $\mathbb{Z}$: basic algorithms for finite abelian groups

When working algorithmically with finite-dimensional algebras over a field, we rely on the vector space structure for our computations (most importantly for solving systems of linear equations). However, when presented with an arbitrary finite ring, we would like to be able to handle the situation regardless of whether it contains a field or not. In the absence of an underlying field, it is the additive group structure of the ring in question that we wish to exploit.

This chapter lays the foundation of everything that succeeds it. At the end of it, we will have built a toolbox for working with finite abelian groups within algorithms. This will allow our later algorithms to have a natural proof-like flow. We will not have to think about the bit operations that go on behind the scenes, and we will talk of algebraic structures, rather than of the strings of integers representing them.

Our algorithms are purposely conceptual. In this way, we aim to concentrate on the structural properties of our objects, rather than rely on seemingly random matrix manipulations that end up giving the "right" result.

We will represent finitely generated abelian groups via generators and relations. Correspondingly, we show how to represent group homomorphisms, subgroups and quotients of groups. Building on this, we describe deterministic polynomial-time algorithms that accomplish the following tasks in the abelian case:

1. test if a group homomorphism is injective,
2. test if a group homomorphism is surjective,
3. decide if two group homomorphisms are equal,
4. compute subgroups generated by a given finite set of elements in a group,

5. compute the quotient of a group by a subgroup,
6. compute kernels, images and cokernels of group homomorphisms,
7. compute direct sums of groups,
8. compute homomorphism groups and tensor products,
9. split exact sequences,
10. compute the order of a finite group,
11. compute the torsion subgroup of a finitely generated group,
12. compute the order of a given group element,
13. compute the exponent of a finite group,
14. write a finitely generated group as a direct sum of cyclic groups.

The last of these is particularly important, as it will allow us to assume in later chapters that a finite abelian group is given by specifying the sizes of its cyclic direct summands.

Working with finitely generated abelian groups in the representation we have chosen ultimately reduces to carrying out integer matrix computations. The way to keep the entries of these matrices under control is either to employ modular techniques, or to give the group a lattice structure and use basis reduction algorithms.

2.1 Lattices

The main references for this section are [67, 68].

Definition 2.1.1. *A* lattice *is an additive subgroup $L \subseteq \mathbb{R}^n$, where $n \in \mathbb{Z}_{\geq 0}$, for which there exists $\epsilon \in \mathbb{R}_{>0}$ such that for all $x \in L$, $x \neq 0$, we have $\langle x, x\rangle \geq \epsilon$, where $\langle \cdot, \cdot \rangle$ denotes the standard inner product on $\mathbb{R}^n$. A* sublattice *of L is a subgroup of L.*

Proposition 2.1.2 ([68], Section 2)**.** *A subset $L \subset \mathbb{R}^n$ is a lattice if and only if there exists a set $B \subset \mathbb{R}^n$ of $\mathbb{R}$-linearly independent vectors such that*

$$L = \sum_{b \in B} \mathbb{Z} b.$$

A set B as in Proposition 2.1.2 is said to be a *basis* of L, and the cardinality of B is the *rank* of L. Suppose $B = \{b_1, \ldots, b_m\}$, for some $m \in \mathbb{Z}_{>0}$ and let A be the matrix whose i^{th} column is given by b_i. The *determinant* of L is

$$\det(L) := \det(\langle b_i, b_j \rangle)_{1 \leq i,j \leq m}^{1/2} = |\det(A)|.$$

It can be shown that the rank and determinant of a lattice are well-defined.

Definition 2.1.3. *Two lattices L and L' are said to be* isomorphic *if there exists a bijective $\mathbb{Z}$-linear transformation $\tau : L \to L'$ such that for all $x, y \in L$, we have $\langle x, y\rangle = \langle \tau(x), \tau(y)\rangle$. If such a transformation exists, we write $L \cong L'$.*

Since most real numbers cannot be represented inside algorithms using a finite number of bits, we will only consider lattices whose vectors are rational numbers. In this case, we represent a lattice by giving a matrix $A \in \mathcal{M}_{n\times m}(\mathbb{Q})$ of rank m. Then L is taken to be the lattice with basis given by the m columns of A, and we write $L = \mathcal{L}(A)$.

An important notion in the theory of lattices is that of a *reduced basis*. A precise definition can be found in [68], Section 10. Intuitively, reduced bases can be thought of as consisting of "short" vectors that are "nearly orthogonal". To the notion of a reduced basis we associate a parameter $c > 4/3$. Roughly speaking, c is a qualitative measure of the reduction – the smaller the value of c, the better the reduction. When no such parameter is specified, it is typically taken to be 2.

An algorithm that, given a lattice, produces a reduced basis thereof is called a *lattice basis reduction algorithm*. An example of such an algorithm, that is deterministic and runs in polynomial time, is the LLL algorithm ([63]).

Definition 2.1.4. *Let L be a lattice of rank n in $\mathbb{R}^n$. The* dual lattice *of L is given by*

$$L^\star = \{x \in \mathbb{R}^n \mid \langle x, L\rangle \subset \mathbb{Z}\},$$

where $\langle\cdot,\cdot\rangle$ is the standard inner product.

Note 2.1.5.

(i) The dual lattice is a lattice.
(ii) $\mathrm{rank}(L^\star) = \mathrm{rank}(L)$ and $\det(L^\star) = \det(L)^{-1}$.
(iii) $L^{\star\star} = L$.
(iv) If L has basis given by the columns of a matrix A, then $L^\star$ has basis given by the columns of the inverse of the transpose of A.

2.1.1 Kernels, images and systems of linear equations over $\mathbb{Z}$

One of the basic tools that we will use is the efficient computability of kernels and images.

Theorem 2.1.6 ([68], Section 14)**.** *There exists a deterministic polynomial-time algorithm that, given a triple (m, n, f), with $n, m \in \mathbb{Z}_{\geq 0}$ and $f \in \mathcal{M}_{n\times m}(\mathbb{Z})$ a matrix representing a group homomorphism $f : \mathbb{Z}^m \to \mathbb{Z}^n$, computes $k := \mathrm{rank}(f)$ and a basis $b_1, \ldots, b_m$ for $\mathbb{Z}^m$ such that $b_1, \ldots, b_{m-k}$ is a basis for* $\ker f$ *and $f(b_{m-k+1}), \ldots, f(b_m)$ is a basis for* $\mathrm{im}\, f$.

This algorithm can then be used to solve systems of linear equations over $\mathbb{Z}$.

Theorem 2.1.7 ([68], Section 14)**.** *There exists a deterministic polynomial-time algorithm that, given a triple (m, n, f), with $n, m \in \mathbb{Z}_{\geq 0}$ and $f \in \mathcal{M}_{n\times m}(\mathbb{Z})$, together with a vector $b \in \mathbb{Z}^n$, computes the set of solutions of the equation $fx = b$, or determines that there is no solution.*

2.1.2 Intersection, sum, inclusion and equality of lattices

A subgroup $H \subseteq \mathbb{Z}^n$ is given to an algorithm by specifying a sequence of elements of $\mathbb{Z}^n$ that is a basis of H over $\mathbb{Z}$. Note that by Theorem 2.1.6, we can recover a basis of H from any generating set.

Proposition 2.1.8. *There exists a deterministic polynomial-time algorithm that, given $n \in \mathbb{Z}_{>0}$ and two subgroups $H_1, H_2 \subseteq \mathbb{Z}^n$, computes $H_1 \cap H_2$ and $H_1 + H_2$, together with the inclusion maps $H_1 \cap H_2 \to H_i$ and $H_i \to H_1 + H_2$, for $i = 1, 2$.*

Proof. Consider the group $H_1 \oplus H_2$, i.e. the group with elements of the form (h_1, h_2), where $h_1 \in H_1$ and $h_2 \in H_2$, together with componentwise addition. Let $\phi : H_1 \oplus H_2 \to \mathbb{Z}^n$ be the map given by $(h_1, h_2) \mapsto h_1 - h_2$. Then $\ker(\phi) = H_1 \cap H_2$ and $\mathrm{im}(\phi) = H_1 + H_2$, and both can be efficiently computed by Theorem 2.1.6. This produces bases of $H_1 \cap H_2$ and $H_1 + H_2$ in terms of the standard basis of $\mathbb{Z}^n$.

Now $H_1 \cap H_2$ is equal to the image of the projection $\ker(\phi) \to H_1$. This gives a basis for $H_1 \cap H_2$ in terms of the basis of H_1. Similarly for H_2. Further, $H_1 = H_1 \cap (H_1 + H_2)$, which gives a basis for H_1 in terms of the basis of $H_1 + H_2$.

□

As a consequence of this, we are able to determine inclusion and equality of two subgroups of $\mathbb{Z}^n$.

Corollary 2.1.9. *There exists a deterministic polynomial-time algorithm such that, given $n \in \mathbb{Z}_{>0}$ and two subgroups $H_1, H_2 \subseteq \mathbb{Z}^n$, determines whether $H_1 \subseteq H_2$.*

Proof. Note that $H_1 \subseteq H_2$ if and only if $H_1 \cap H_2 = H_1$. Since $H_1 \cap H_2 \subseteq H_1$, testing equality is equivalent to testing whether the determinants of the two lattices, $H_1 \cap H_2$ and H_1, are equal. Computing determinants of lattices reduces to computing determinants of integer matrices, which can be done in polynomial time. □

Corollary 2.1.10. *There exists a deterministic polynomial-time algorithm such that, given $n \in \mathbb{Z}_{>0}$ and two subgroups $H_1, H_2 \subseteq \mathbb{Z}^n$, determines whether $H_1 = H_2$.*

2.2 Hermite and Smith normal forms

This section draws on Section 2.4 of [19].

There are two canonical forms of a matrix A that are of interest: the *Hermite normal form* and the *Smith normal form*. These can be obtained by applying row and column operations to A.

Definition 2.2.1. *Let $m, n \in \mathbb{Z}_{>0}$ and $A \in \mathcal{M}_{n \times m}(\mathbb{Z})$. A* column operation *on A is one of the following:*

(i) *interchanging two columns of A,*
(ii) *multiplying one column of A by -1,*
(iii) *adding a nonzero multiple of a column of A to another column.*

Note 2.2.2. (i) Each column operation corresponds to postmultiplying A with an appropriate invertible matrix over $\mathbb{Z}$.
(ii) If A' is a matrix obtained from A via a sequence of column operations, then there exists an invertible matrix V such that $A' = AV$. Conversely, if two matrices differ by a postmultiplied invertible matrix, then one can be obtained from the other by a series of column operations.
(iii) Applying column operations to a square matrix does not change the absolute value of its determinant.

Note 2.2.3. We can similarly define *row operations*. These correspond to premultiplying by a certain invertible matrix over $\mathbb{Z}$.

It is easy to see that performing column operations on a matrix does not change the lattice the columns generate.

Proposition 2.2.4. *Let $A, B \in \mathcal{M}_{n\times m}(\mathbb{Z})$. Then the lattice generated by the columns of A is equal to the lattice generated by the columns of B if and only if there exists $V \in \mathrm{GL}_m(\mathbb{Z})$ such that $AV = B$.*

Note 2.2.5. Let F be a free $\mathbb{Z}$-module of finite rank. In choosing to represent a subgroup $H \hookrightarrow F$ via a matrix $A \in \mathcal{M}_{n\times m}(\mathbb{Z})$, we are making a choice of basis of H and of F. Applying column operations to A corresponds to a change of basis of H, while keeping the basis for F fixed. Applying row operations corresponds to a change of basis of F, while keeping the basis for H fixed.

We are now ready to introduce the Hermite normal form, which is useful for representing subgroups of $\mathbb{Z}^n$ in a canonical way.

Definition 2.2.6. *Let $A = (a_{i,j}) \in \mathcal{M}_{n\times m}(\mathbb{Z})$, for some $m, n \in \mathbb{Z}_{>0}$. Then A is said to be in* Hermite normal form (HNF) *if there exists $0 \leq k \leq m$ such that the last $m - k$ columns are zero and for each $1 \leq j \leq k$, there exists an entry $a_{i_j,j} > 0$ such that*

(i) *For all $i' < i_j$, we have $a_{i',j} = 0$.*
(ii) *For all $j' < j$, we have $a_{i_j,j} > a_{i_j,j'} \geq 0$.*
(iii) *For all $j' < j$, we have $i_{j'} < i_j$.*

Note 2.2.7. The nonzero entry $a_{i_j,j}$ is called the *leading coefficient* of the j^{th} column. Informally, a matrix is in Hermite normal form if all its zero columns lie on the right, the leading coefficients of all nonzero columns are strictly positive and have nonnegative and strictly smaller entries to their left, and occur strictly below the position of the leading coefficient of the previous column, if this exists.

Note 2.2.8. We have seen that applying column operations to a matrix does not change the lattice it generates. Thus, finding the Hermite normal form of a matrix corresponds to finding a basis of the associated lattice, such that the basis vectors can be ordered in such a way that they have an increasing number of leading zero entries.

Proposition 2.2.9. (i) *Each integer matrix can transformed into a matrix in Hermite normal form by a sequence of column operations.*

(ii) ([12], Section 5.3) *Each integer matrix has a unique Hermite normal form, i.e. if $A = (a_{i,j}) \in \mathcal{M}_{n\times m}(\mathbb{Z})$, then there exists a matrix $V \in \mathrm{GL}_m(\mathbb{Z})$ such that AV is in Hermite normal form. If there is another $V' \in \mathrm{GL}_m(\mathbb{Z})$ such that AV' is in Hermite normal form, then $AV = AV'$.*

Sketch of proof of (i). Let $A = (a_{i,j}) \in \mathcal{M}_{n\times m}(\mathbb{Z})$. Suppose that A has block form

$$A = \begin{bmatrix} A_0 & \mathbf{0} \end{bmatrix},$$

where A_0 is an $n \times m'$ matrix, for some $m' \leq m$, with no nonzero columns. Otherwise interchange columns to arrive at this form. Let $a_{01}, \ldots, a_{0m'}$ be the entries in the first nonzero row of A_0. Then at least one a_{0i} must be nonzero and we can ensure that they are all nonnegative (by applying suitable column operations), so using the extended Euclid algorithm (see [33]), we can compute $g := \gcd(\{a_{0i}\}_i)$. This reduces to applying a sequence of column operations at the end of which the first nonzero row of A will be $[g \;\; 0 \;\; 0 \ldots 0]$. The matrix now has form

$$\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ g & 0 & 0 \\ \star & A_1 & \mathbf{0} \end{bmatrix}.$$

We proceed by computing the greatest common divisor of the entries in the first nonzero row of A_1. We perform a couple of extra column operations to ensure that the entry to the left of the leading entry of the second column is strictly smaller than the leading entry. We continue in this way until we attain the Hermite normal form. □

Note 2.2.10. The matrix V in Proposition 2.2.9, part (ii), need not be unique.

Note 2.2.11. Since the Hermite normal form is unique, we see that the leading entry of the first column is the greatest common divisor of the entries in the first nonzero row of the original matrix.

There are many algorithms available in the literature that compute the Hermite normal form of a given integer matrix. The main difficulty in achieving polynomial time is to keep the entries of the intermediate matrices small. The straightforward column-operation-algorithm presented in Proposition 2.2.9 suffers from coefficient blow-up. This can be avoided by using modular techniques, for example by working modulo an integer d, where d is chosen to be the determinant of a full rank submatrix of A. Another way to circumvent coefficient blow-up is to employ lattice basis reduction techniques.

For detailed accounts of deterministic polynomial-time algorithms for computing the Hermite normal form of an integer matrix, together with a transformation matrix V, see [16, 30, 33, 85]. We will only record their existence.

Theorem 2.2.12 ([12], Proposition 5.4)**.** *There exists a deterministic polynomial-time algorithm that, given a matrix $A = (a_{i,j}) \in \mathcal{M}_{n\times m}(\mathbb{Z})$, computes a matrix $V \in \mathrm{GL}_m(\mathbb{Z})$ such that AV is in Hermite normal form.*

Note 2.2.13. Theorem 2.2.12 can also be obtained from Theorem 2.1.6 if we equip $\mathcal{L}(A)$ with a suitable "length function" $q : L \to \mathbb{R}$ (see [68], Section 14).

The second canonical form we wish to examine is the Smith normal form.

Definition 2.2.14. *Let $A = (a_{i,j}) \in \mathcal{M}_{n\times m}(\mathbb{Z})$, for some $m, n \in \mathbb{Z}_{\geq 0}$. Then A is said to be in* Smith normal form (SNF) *if there exists $0 \leq k \leq \min\{n, m\}$ such that the last $n-k$ rows and the last $m-k$ columns are zero and the matrix $(a_{i,j})_{i=1,j=1}^{k,k}$ is diagonal, with $a_{i,i} > 0$ for all $1 \leq i \leq k$, and $a_{i,i} \mid a_{i+1,i+1}$ for all $1 \leq i < k$.*

Note 2.2.15. The nonzero entries are called *elementary divisors* of A.

The following is a standard result.

Proposition 2.2.16. *Every integer matrix has a unique Smith normal form, i.e. if $A = (a_{i,j}) \in \mathcal{M}_{n\times m}(\mathbb{Z})$, then there exist matrices $U \in \mathrm{GL}_n(\mathbb{Z})$ and $V \in \mathrm{GL}_m(\mathbb{Z})$ such that UAV is in Smith normal form. If there exist other $U' \in \mathrm{GL}_n(\mathbb{Z})$ and $V' \in \mathrm{GL}_m(\mathbb{Z})$ such that $U'AV'$ is in Smith normal form, then $U'AV' = UAV$.*

Theorem 2.2.17 ([86], Section 8.2)**.** *There exists a deterministic polynomial-time algorithm that, given a matrix $A = (a_{i,j}) \in \mathcal{M}_{n\times m}(\mathbb{Z})$, computes the Smith normal form of A, together with transformation matrices $U \in \mathrm{GL}_n(\mathbb{Z})$ and $V \in \mathrm{GL}_m(\mathbb{Z})$.*

The relevance of the Hermite and Smith normal forms to the study of finitely generated abelian groups lies in the following theorem.

Theorem 2.2.18. *Let $n \in \mathbb{Z}_{>0}$. Suppose $H \subseteq \mathbb{Z}^n$ is a subgroup.*

(i) *There exists a unique full column rank matrix A in Hermite normal form such that H is generated over $\mathbb{Z}$ by the columns of A.*
(ii) *If H has rank n, then there exists a unique square matrix A in Smith normal form such that*

$$\mathbb{Z}^n/H \cong \bigoplus_{i=1}^{n} \mathbb{Z}/d_i\mathbb{Z}, \tag{2.1}$$

as $\mathbb{Z}$-modules, where $d_1, \ldots, d_n$ are the elementary divisors of A.

Proof. Part (i) is a consequence of Propositions 2.2.4 and 2.2.9. For a proof of part (ii), see Theorem 2.4.13 of [19]. □

2.3 Representing objects and basic constructions

Given that we wish to bound the running time of an algorithm in terms of the length of the input, it is of crucial importance to make clear how we represent objects inside algorithms. Different representations may lead to essentially different computational tasks, with different complexities.

There are several ways to represent groups inside an algorithm. These include giving a finite presentation, giving the group as a permutation group of a finite set or a matrix group over a ring, black-box representations, or giving the group as automorphisms of certain objects (e.g. graphs, field extensions). For more details on group representations and the algorithmic problems they give rise to, see [35, 38].

For finitely generated abelian groups, matters simplify greatly, since these are nothing else than $\mathbb{Z}$-modules and thus can be represented by matrices.

2.3.1 Representing finite and finitely generated abelian groups

The proof of the following result can be found in any introductory algebra textbook.

Theorem 2.3.1. *Let G be a finitely generated abelian group. Then:*

(i) *There exists $k \in \mathbb{Z}_{\geq 0}$ such that $G \cong \mathbb{Z}^k \oplus H$, where H is a finite abelian group.*
(ii) *If G is finite, then there exist $n \in \mathbb{Z}_{>0}$ and a subgroup $L \subseteq \mathbb{Z}^n$ of rank n, such that $G \cong \mathbb{Z}^n/L$.*
(iii) (Fundamental Theorem of Finite Abelian Groups) *If G is finite, then there exists a unique $t \in \mathbb{Z}_{\geq 0}$ and a unique sequence of integers $d_1, \ldots, d_t \in \mathbb{Z}_{>1}$ such that $d_1 \mid d_2 \mid \ldots \mid d_t$ and*

$$G \cong \bigoplus_{i=1}^{t} \mathbb{Z}/d_i\mathbb{Z}.$$

To represent a group G, we give the algorithm a set of generators and relations. More precisely, suppose G has generators $x_1, \ldots, x_n$ and relations in G are of the form $\sum_{i=1}^{n} a_{ij}x_i$ for $1 \leq j \leq m$, where $a_{ij} \in \mathbb{Z}$ for all i, j. Then the matrix $f = (a_{ij}) \in \mathcal{M}_{n\times m}(\mathbb{Z})$ is said to be a *presentation matrix* of G. Consider the exact sequence

$$\mathbb{Z}^m \xrightarrow{f} \mathbb{Z}^n \to \operatorname{coker}(f) \to 0,$$

Then $\operatorname{coker}(f) \cong G$ and an element $g \in \operatorname{coker}(f)$ corresponds, in a non-unique way, to a vector in $\mathbb{Z}^n$ mapping to g, i.e. it is specified as a $\mathbb{Z}$-linear combination of generators.

Definition 2.3.2. *Let G be a finitely generated abelian group. An* exact-sequence representation *of G consists of a triple (m, n, f), where $m, n \in \mathbb{Z}_{>0}$ and $f \in \operatorname{Hom}(\mathbb{Z}^m, \mathbb{Z}^n)$ are such that $\operatorname{coker}(f) \cong G$.*

Proposition 2.3.3. *Let G be a finitely generated abelian group and let $f \in \mathcal{M}_{n\times m}(\mathbb{Z})$ be a presentation matrix of G. If $n = m$, then G is finite if and only if $\det(f) \neq 0$. In this case, $|G| = |\det(f)|$.*

Proof. This follows from Theorem 2.2.18, Theorem 2.3.1. □

From now on, given a presentation matrix f for a finitely generated abelian group G, we will identify G with $\operatorname{coker}(f)$.

2.3.2 Group homomorphisms

Let G_1, G_2 be two finitely generated abelian groups. Suppose G_1 and G_2 are represented by triples (m_1, n_1, f_1) and (m_2, n_2, f_2) respectively. Then we have two exact sequences:

$$\begin{array}{ccccccc} \mathbb{Z}^{m_1} & \xrightarrow{f_1} & \mathbb{Z}^{n_1} & \xrightarrow{\pi_1} & G_1 & \longrightarrow & 0 \\ & & \downarrow & & \downarrow & & \\ \mathbb{Z}^{m_2} & \xrightarrow{f_2} & \mathbb{Z}^{n_2} & \xrightarrow{\pi_2} & G_2 & \longrightarrow & 0. \end{array}$$

Any group homomorphism $G_1 \to G_2$ is induced by a map $\mathbb{Z}^{n_1} \to \mathbb{Z}^{n_2}$. This is the same as saying that a group homomorphism is determined by the images of the generators. However, not all assignments of generators correspond to well-defined group homomorphisms. That is to say, not every map $g : \mathbb{Z}^{n_1} \to \mathbb{Z}^{n_2}$ gives rise to a group homomorphism $\overline{g} : G_1 \to G_2$.

Proposition 2.3.4. *There exists a deterministic polynomial-time algorithm that, given two finitely generated abelian groups G_1 and G_2, represented by triples (m_1, n_1, f_1) and (m_2, n_2, f_2) respectively, and a map $g : \mathbb{Z}^{n_1} \to \mathbb{Z}^{n_2}$, decides whether or not g induces a group homomorphism $\overline{g} : G_1 \to G_2$.*

Proof. The map induced by g takes an element of G_1, lifts it to $\mathbb{Z}^{n_1}$ and then maps it to G_2 under $\pi_2 \circ g$. For this to be a well-defined group homomorphism, we must ensure that it is independent of the lift to $\mathbb{Z}^{n_1}$ we choose. For this, we require that $\operatorname{im}(f_1) \subseteq \ker(\pi_2 \circ g)$, or equivalently, $\operatorname{im}(g \circ f_1) \subseteq \operatorname{im}(f_2)$.

By Theorem 2.1.6 and Corollary 2.1.9, we can compute kernels and images, and test for inclusion, so, given a map $g : \mathbb{Z}^{n_1} \to \mathbb{Z}^{n_2}$, we can test if g induces a group homomorphism $G_1 \to G_2$ by checking whether

$$\operatorname{im}(g \circ f_1) \subseteq \operatorname{im}(f_2).$$

□

Note 2.3.5. The condition $\operatorname{im}(g \circ f_1) \subseteq \operatorname{im}(f_2)$ is equivalent to the existence of a map $h : \mathbb{Z}^{m_1} \to \mathbb{Z}^{m_2}$ such that $g \circ f_1 = f_2 \circ h$.

Once we have ensured that we have a well-defined group homomorphism, we may test whether it is injective or surjective.

Proposition 2.3.6. *There exists a deterministic polynomial-time algorithm that, given two finitely generated abelian groups G_1 and G_2, represented by triples (m_1, n_1, f_1) and (m_2, n_2, f_2) respectively, and a map $g : \mathbb{Z}^{n_1} \to \mathbb{Z}^{n_2}$ inducing a group homomorphism $\overline{g} : G_1 \to G_2$, decides whether $\overline{g}$ is injective.*

Proof. For $\overline{g}$ to be injective, we require that for all $x \in \mathbb{Z}^{n_1}$, if $g(x) \in \mathrm{im}(f_2)$, then $x \in \mathrm{im}(f_1)$, or equivalently, that

$$\mathrm{im}(f_1) \supseteq g^{-1}(\mathrm{im}(f_2)).$$

To express this condition in terms of the maps that are part of the input, i.e. in terms of f_1, f_2 and g, consider the map

$$-f_2 + g : \mathbb{Z}^{m_2} \oplus \mathbb{Z}^{n_1} \to \mathbb{Z}^{n_2}, \quad (x, y) \mapsto -f_2(x) + g(y). \tag{2.2}$$

Let $p_2 : \mathbb{Z}^{m_2} \oplus \mathbb{Z}^{n_1} \to \mathbb{Z}^{n_1}$ be the projection map to the second component. It is easy to see that

$$g^{-1}(\mathrm{im}(f_2)) = p_2(\ker(-f_2 + g)).$$

Thus, the condition for injectivity becomes

$$p_2(\ker(-f_2 + g)) \subseteq \mathrm{im}\, f_1,$$

which can be tested deterministically in polynomial time using Theorem 2.1.6 and Corollary 2.1.9. □

Proposition 2.3.7. *There exists a deterministic polynomial-time algorithm that, given two finitely generated abelian groups G_1 and G_2, represented by triples (m_1, n_1, f_1) and (m_2, n_2, f_2) respectively, and a map $g : \mathbb{Z}^{n_1} \to \mathbb{Z}^{n_2}$ inducing a group homomorphism $\overline{g} : G_1 \to G_2$, decides whether $\overline{g}$ is surjective.*

Proof. For $\overline{g} : G_1 \to G_2$ to be surjective, we require that for all $\overline{x} \in G_2 = \mathbb{Z}^{n_2}/\mathrm{im}(f_2)$, there exist $y \in \mathbb{Z}^{n_1}$ such that $\pi_2 \circ g(y) = \overline{x}$. This is equivalent to requiring that

$$\forall x \in \mathbb{Z}^{n_2}, \quad \exists y \in \mathbb{Z}^{n_1} \text{ such that } g(y) - x \in \mathrm{im}(f_2),$$

which is further equivalent to the map $-f_2 + g$ defined in (2.2), being surjective. We can check this using Theorem 2.1.6. □

Proposition 2.3.8. *There exists a deterministic polynomial-time algorithm that, given two finitely generated abelian groups G_1 and G_2, represented by triples (m_1, n_1, f_1) and (m_2, n_2, f_2) respectively, and two maps $g, h : \mathbb{Z}^{n_1} \to \mathbb{Z}^{n_2}$ inducing group homomorphisms $\overline{g}, \overline{h} : G_1 \to G_2$, decides whether $\overline{g} = \overline{h}$.*

Proof. The problem of deciding if two group homomorphisms are equal is equivalent to deciding if a given group homomorphism is the zero homomorphism. Let $\overline{g} : G_1 \to G_2$ be a group homomorphism induced by the map $g : \mathbb{Z}^{n_1} \to \mathbb{Z}^{n_2}$. Then $\overline{g} \equiv 0$ if and only if $\mathrm{im}(g) \subseteq \mathrm{im}(f_2)$. □

2.3.3 Subgroups and quotients

Representing subgroups

Let G be a finitely generated abelian group. To represent a subgroup H of G, we need to represent it as a group and additionally, to produce an embedding $H \hookrightarrow G$ that tells us how H sits inside G, i.e. we need to give the algorithm a triple (r, s, h) and a map of free $\mathbb{Z}$-modules g, that induces an injective map $H \hookrightarrow G$.

Example 2.3.9. It is important to specify the injection: if $G = \mathbb{Z}/2\mathbb{Z} \oplus \mathbb{Z}/2\mathbb{Z}$, then the subgroup $\mathbb{Z}/2\mathbb{Z}$ sits inside G in three different ways.

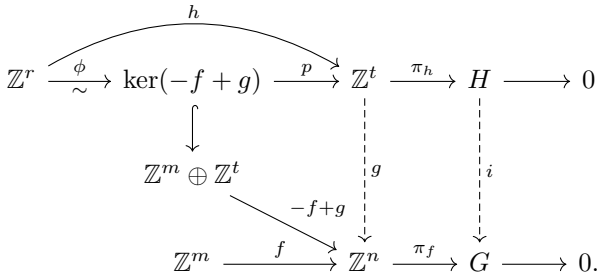
Proposition 2.3.10. *There exists a deterministic polynomial-time algorithm that, given a finitely generated abelian group G and a finite set $\overline{S} \subset G$, computes the subgroup $H \leq G$ generated by $\overline{S}$.*

Proof. The desired output of the algorithm is a triple (r, s, h) representing H as a group, together with an injective group homomorphism $i : H \hookrightarrow G$. Suppose the group G is represented by triple (m, n, f).

Let $S = \{s_1, \ldots, s_t\} \subset \mathbb{Z}^n$ be a set of representatives of the elements of $\overline{S}$ in G. Consider the map $g : \mathbb{Z}^t \to \mathbb{Z}^n$ given by $e_i \mapsto s_i$, where for $1 \leq i \leq t$, the vector e_i is the i^{th} canonical basis element of $\mathbb{Z}^t$:

$$\begin{array}{ccccccc} & & & & \mathbb{Z}^t & & \\ & & & & \downarrow{\scriptstyle g} & & \\ \mathbb{Z}^m & \xrightarrow{f} & \mathbb{Z}^n & \xrightarrow{\pi_f} & G & \longrightarrow & 0. \end{array} \tag{2.3}$$

Consider the map $(-f + g) : \mathbb{Z}^m \oplus \mathbb{Z}^t \to \mathbb{Z}^n$, given by $(x, y) \mapsto -f(x) + g(y)$. Compute $\ker(-f + g) = \{(x, y) \in \mathbb{Z}^m \oplus \mathbb{Z}^t \mid f(x) = g(y)\}$ and project it to $\mathbb{Z}^t$ via a map p. Put $r := \operatorname{rank}(\ker(-f + g))$ and let ϕ be the matrix whose columns are the basis vectors of $\ker(-f + g)$. The following diagram illustrates what was described above:



Put $h := p \circ \phi$. Then (r, t, h) represents H. Note that t is not necessarily the minimum number of generators of H.

The map g induces a map $i : H \to G$ in the following way: an element of $H = \mathbb{Z}^t / \operatorname{im}(h)$ is lifted to $\mathbb{Z}^t$ and then is mapped to G. We claim that $i : H \to G$ is an injective group homomorphism.

First we show that i is independent of the chosen lift to $\mathbb{Z}^t$. Let $\overline{x} \in H$ and lift it to $x+k \in \mathbb{Z}^t$, for some $k \in \mathrm{im}(h)$. Then $g(x+k) = g(x)+g(k)$. But, since $k \in \mathrm{im}(h)$, it follows that k is the image under p of some $(y,k) \in \ker(-f+g)$, with $y \in \mathbb{Z}^m$. Then $g(k) = f(y)$. Hence $\pi_f g(k) = 0$ and so $\overline{g}$ does not depend on the lift.

For injectivity, suppose $i(\overline{y}) = 0_G$, for some $\overline{y} \in H$, where 0_G denotes the zero element in G. Let y be a lift of $\overline{y}$ to $\mathbb{Z}^t$. Then $\pi_f(g(y)) = 0_G$, so $g(y) = f(x)$, for some $x \in \mathbb{Z}^t$. But then $(x,y) \in \ker(-f+g)$ and so $y \in \mathrm{im}(p)$, i.e. $\overline{y} = 0_H$.

Moreover, by construction, $\mathrm{im}(i) = \langle\{\overline{s_i}\}\rangle = H$, where $\overline{s_i} := \pi_f(s_i)$. □

Example 2.3.11. Let G be any group. If $S = \emptyset$, then $t = 0$, and $H = 0$, as expected.

Example 2.3.12. Let $G = \mathbb{Z}/2\mathbb{Z} \oplus \mathbb{Z}/4\mathbb{Z} \oplus \mathbb{Z}/4\mathbb{Z}$.

(i) Let $S = \{(1,1,2)\}$. Then $H := \langle S\rangle \cong \mathbb{Z}/4\mathbb{Z}$. The map $g - f : \mathbb{Z}^3 \oplus \mathbb{Z} \to \mathbb{Z}^3$ is given by $(x_1,x_2,x_3,y) \mapsto (y-2x_1, y-4x_2, 2y-4x_3)$, so $\ker(g-f) = (2,1,2,4)\mathbb{Z}$. Hence $\mathrm{im}(\pi) = 4\mathbb{Z}$ and the subgroup H is given by $(1,1,4\cdot\mathrm{id})$.

(ii) Let $S = \{(1,1,2),(0,2,0)\}$. Again, $H := \langle S\rangle \cong \mathbb{Z}/4\mathbb{Z}$. The map $g-f : \mathbb{Z}^3 \oplus \mathbb{Z}^2 \to \mathbb{Z}^3$ is given by $(x_1,x_2,x_3,y_1,y_2) \mapsto (2y_1 - 4x_1, y_1 + 2y_2 - 4x_2, 2y_1 - 4x_3)$, so $\ker(g-f) = (1,0,1,2,1)\mathbb{Z} \oplus (0,1,0,0,2)\mathbb{Z}$. Hence $\mathrm{im}(\pi) = (2,1)\mathbb{Z} \oplus (0,2)\mathbb{Z}$ and the subgroup H is given by $(2,2,h)$, where h is the map represented by the matrix

$$\begin{pmatrix} 2 & 1 \\ 0 & 2 \end{pmatrix}.$$

It can be checked that the Smith normal form of this matrix has diagonal $(4,1)$, so $(2,2,h)$ indeed represents the cyclic group of order 4.

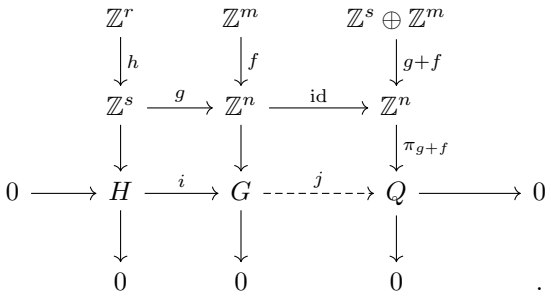
Representing quotients

Let G be a finitely generated abelian group and let $H \leq G$ be a subgroup. To represent the quotient G/H, we need to represent it as a group, which we denote by Q, and additionally, to produce a surjection $j : G \twoheadrightarrow Q$. Then $Q \cong G/H$.

Proposition 2.3.13. *There exists a deterministic polynomial-time algorithm that, given a finitely generated abelian group G and a subgroup $H \leq G$, computes the quotient group G/H.*

Proof. Suppose the group G is represented by triple (m,n,f) and H is represented by triple (r,s,h), together with a map $g : \mathbb{Z}^s \to \mathbb{Z}^n$ inducing an injection $i : H \to G$, as in the previous subsection. Let $g+f : \mathbb{Z}^s \oplus \mathbb{Z}^m \to \mathbb{Z}^n$ be the map given by

$(x, y) \mapsto g(x) + f(y)$. Consider the following diagram

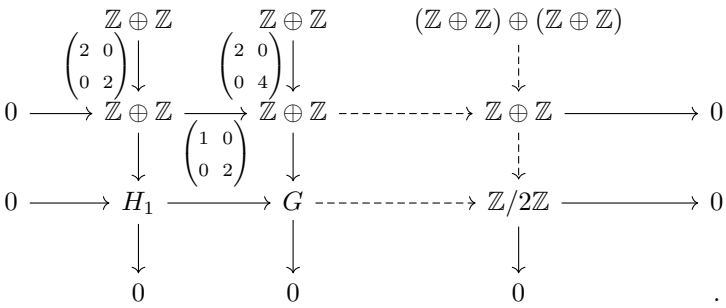


Then G/H is represented by triple $(s + m, n, g + f)$.

The map $g + f$ induces a map $j : G \to Q$ in the following way: an element of $G = \mathbb{Z}^n/\operatorname{im}(f)$ is lifted to $\mathbb{Z}^n$ and then mapped to Q via $\pi_{g+f} \circ \mathrm{id}$. We claim that this map is a surjective group homomorphism.

To see that it is well-defined, let $\overline{x} \in G$ and a consider a lift of $\overline{x}$ to $\mathbb{Z}^n$ given by $x + f(y)$, for some $y \in \mathbb{Z}^m$. Since $f(y) \in \operatorname{im}(g + f)$ by construction, it is sent to zero under π_{g+f}. Hence j is independent of the choice of lift. Surjectivity follows by construction. □

Example 2.3.14. Let $G = \mathbb{Z}/2\mathbb{Z} \oplus \mathbb{Z}/4\mathbb{Z}$. Let $H_1 = \mathbb{Z}/2\mathbb{Z} \oplus \mathbb{Z}/2\mathbb{Z}$. Then the quotient G/H_1 is given by data

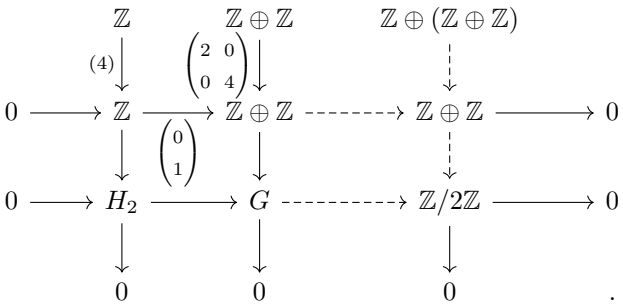


The presentation matrix for G/H_1 is

$$f_{G/H_1} = \begin{pmatrix} 1 & 0 & 2 & 0 \\ 0 & 2 & 0 & 4 \end{pmatrix},$$

whose Smith normal form has nonzero entries $(2\ \ 1)$, and so $G/H_1 \cong \mathbb{Z}/2\mathbb{Z}$.

If we let $H_2 = \mathbb{Z}/4\mathbb{Z}$, the quotient G/H_2 is given by data



Then the presentation matrix for G/H_2 is

$$f_{G/H_2} = \begin{pmatrix} 0 & 2 & 0 \\ 1 & 0 & 4 \end{pmatrix},$$

whose Smith normal form again has nonzero entries $(2\ 1)$, and so $G/H_2 \cong \mathbb{Z}/2\mathbb{Z}$.

Proposition 2.3.15. *There exists a deterministic polynomial-time algorithm that, given two finitely generated abelian groups G_1 and G_2, represented by triples (m_1, n_1, f_1) and (m_2, n_2, f_2) respectively, and a map $g : \mathbb{Z}^{n_1} \to \mathbb{Z}^{n_2}$, computes*

(i) $\ker(\overline{g})$, *together with the corresponding injective group homomorphism* $\ker(\overline{g}) \to G_1$,
(ii) $\operatorname{im}(\overline{g})$, *together with the corresponding injective group homomorphism* $\operatorname{im}(\overline{g}) \to G_2$,
(iii) $\operatorname{coker}(\overline{g})$, *together with the corresponding surjective group homomorphism* $G_2 \to \operatorname{coker}(\overline{g})$,

where $\overline{g} : G_1 \to G_2$ is the group homomorphism induced by g.

Proof. Kernels can be computed as:

$$\begin{aligned} \ker(G_1 \to G_2) &= \ker\left(\mathbb{Z}^{n_1}/\operatorname{im}(f_1) \longrightarrow \mathbb{Z}^{n_2}/\operatorname{im}(f_2)\right) \\ &= \{x \in \mathbb{Z}^{n_1} \mid \pi_2 \circ g(x) = 0\}/\operatorname{im}(f_1) \\ &= \ker(\pi_2 \circ g)/\operatorname{im}(f_1) \\ &= p_2(\ker(-f_2 + g))/\operatorname{im}(f_1). \end{aligned}$$

Images can be computed as:

$$\begin{aligned} \operatorname{im}(G_1 \to G_2) &= \operatorname{im}(\pi_2 \circ g) \\ &= \operatorname{im}(g)/(\operatorname{im}(g) \cap \operatorname{im}(f_2)) \\ &= \operatorname{im}(-f_2 + g)/\operatorname{im}(f_2). \end{aligned}$$

Cokernels can be computed as:

$$\begin{aligned}\operatorname{coker}(G_1 \to G_2) &= G_2/\operatorname{im}(G_1 \to G_2)\\ &= (\mathbb{Z}^{n_2}/\operatorname{im}(f_2))\,/\,(\operatorname{im}(-f_2+g)/\operatorname{im}(f_2))\\ &= \operatorname{coker}(-f_2+g).\end{aligned}$$

All these computations can be carried out in polynomial time by Theorem 2.1.6 and Proposition 2.1.8. The maps $\ker(\overline{g}) \to G_1$, $\operatorname{im}(\overline{g}) \to G_2$ and $G_2 \to \operatorname{coker}(\overline{g})$ can be obtained from Propositions 2.3.10 and 2.3.13. □

2.3.4 Direct sums of groups

Proposition 2.3.16. *There exists a deterministic polynomial-time algorithm that, given two finitely generated abelian group G_1, G_2, computes the direct sum $G_1 \oplus G_2$.*

Proof. Suppose G_1 and G_2 are represented via triples (m_1, n_1, f_1) and (m_2, n_2, f_2) respectively. Then the direct sum $G_1 \oplus G_2$ is represented by triple (m_1+m_2, n_1+n_2, F), where F is an $(m_1+m_2) \times (n_1+n_2)$ integer matrix with block form

$$F = \begin{pmatrix} f_1 & 0 \\ 0 & f_2 \end{pmatrix}.$$

□

2.4 Homomorphism groups and tensor products

In the case of free abelian groups of finite rank, the tensor product is easy to construct: if $\{x_i\}_{i=1}^n$ is a basis for $\mathbb{Z}^n$ and $\{y_i\}_{i=1}^m$ is a basis for $\mathbb{Z}^m$, then the set $\{x_i \otimes y_j \mid 1 \le i,j \le n\}$ is a basis for $\mathbb{Z}^n \otimes \mathbb{Z}^m$. Constructing homomorphism groups of free abelian groups of finite rank is also easy, since $\operatorname{Hom}(\mathbb{Z}^m, \mathbb{Z}^n) = \mathcal{M}_{n\times m}(\mathbb{Z})$.

Suppose now that G_1, G_2 are two finitely generated abelian groups with presentations

$$\mathbb{Z}^{m_1} \xrightarrow{f_1} \mathbb{Z}^{n_1} \longrightarrow G_1 \longrightarrow 0 \tag{2.4}$$

and

$$\mathbb{Z}^{m_2} \xrightarrow{f_2} \mathbb{Z}^{n_2} \longrightarrow G_2 \longrightarrow 0 \tag{2.5}$$

respectively.

Proposition 2.4.1. *There exists a deterministic polynomial-time algorithm that, given two finitely generated abelian groups G_1, G_2, computes*

(i) *the tensor product, $G_1 \otimes G_2$, together with the corresponding bilinear map $G_1 \times G_2 \to G_1 \otimes G_2$,*
(ii) *the homomorphism group, $\operatorname{Hom}(G_1, G_2)$, together with the corresponding bilinear map $\operatorname{Hom}(G_1, G_2) \times G_1 \to G_2$.*

Proof. Suppose G_1, G_2 have presentations (2.4) and (2.5), respectively.

Recall that tensoring is right-exact, so tensoring (2.4) with G_2 and tensoring (2.5) with $\mathbb{Z}^{m_1}$ and $\mathbb{Z}^{n_1}$ gives

$$\begin{array}{ccccccc}
\mathbb{Z}^{m_1}\otimes\mathbb{Z}^{m_2} & & \mathbb{Z}^{n_1}\otimes\mathbb{Z}^{m_2} & & & & \\
\downarrow & & \downarrow & & & & \\
\mathbb{Z}^{m_1}\otimes\mathbb{Z}^{n_2} & & \mathbb{Z}^{n_1}\otimes\mathbb{Z}^{n_2} & & & & \\
\downarrow & & \downarrow & & & & \\
\mathbb{Z}^{m_1}\otimes G_2 & \longrightarrow & \mathbb{Z}^{n_1}\otimes G_2 & \longrightarrow & G_1\otimes G_2 & \longrightarrow & 0. \\
\downarrow & & \downarrow & & & & \\
0 & & 0 & & & &
\end{array}$$

By construction of quotient groups (Theorem 2.3.13), it follows that $G_1 \otimes G_2$ has presentation given by

$$(\mathbb{Z}^{m_1}\otimes\mathbb{Z}^{n_2}) \oplus (\mathbb{Z}^{n_1}\otimes\mathbb{Z}^{m_2}) \longrightarrow \mathbb{Z}^{n_1}\otimes\mathbb{Z}^{n_2} \longrightarrow G_1\otimes G_2 \longrightarrow 0,$$

where the first map is given by $(f_1 \otimes \mathrm{id}) + (\mathrm{id} \otimes f_2)$.

Applying $\mathrm{Hom}(-, G_2)$ to (2.4) gives

$$0 \longrightarrow \mathrm{Hom}(G_1, G_2) \longrightarrow \mathrm{Hom}(\mathbb{Z}^{n_1}, G_2) \longrightarrow \mathrm{Hom}(\mathbb{Z}^{m_1}, G_2).$$

Hence

$$\mathrm{Hom}(G_1, G_2) = \ker\left(\mathrm{Hom}(\mathbb{Z}^{n_1}, G_2) \xrightarrow{\circ f_1} \mathrm{Hom}(\mathbb{Z}^{m_1}, G_2), h \mapsto h \circ f_1\right),$$

which can be computed using Proposition 2.3.15. Now, for $k \in \mathbb{Z}_{\geq 0}$, we have that $\mathrm{Hom}(\mathbb{Z}^k, -)$ is an exact functor, so

$$\mathrm{Hom}(\mathbb{Z}^k, G_2) = \mathrm{coker}\left(\mathrm{Hom}(\mathbb{Z}^k, \mathbb{Z}^{m_2}) \xrightarrow{f_2\circ} \mathrm{Hom}(\mathbb{Z}^k, \mathbb{Z}^{n_2}), h \mapsto f_2 \circ h\right),$$

which we compute for k equal to n_1 and m_1.

The bilinear maps $G_1 \times G_2 \to G_1 \otimes G_2$ and $\mathrm{Hom}(G_1, G_2) \times G_1 \to G_2$ are represented by listing the images of all pairs of generators, and can be readily obtained from the above construction. □

2.5 Splitting exact sequences

Consider an exact sequence of finitely generated abelian groups

$$0 \longrightarrow H \xrightarrow{i} G \xrightarrow{p} K \longrightarrow 0. \tag{2.6}$$

Proposition 2.5.1. *There exists a deterministic polynomial-time algorithm that, given an exact sequence as in (2.6), decides if it is split, and if it is split, produces a right-inverse of p and a left-inverse of i.*

Proof. We consider the map

$$\psi : \mathrm{Hom}(K, G) \to \mathrm{Hom}(K, K), \quad k \mapsto p \circ k.$$

Deciding if the sequence is split reduces to deciding if id_K is in the image of ψ. This can be done by solving a system of linear equations over $\mathbb{Z}$ (Theorem 2.1.7). If the system has a solution, solving it will also produces a right inverse of p, i.e. an element $s \in \mathrm{Hom}(K, G)$ such that $\psi(s) = \mathrm{id}_K$. Similarly, we construct a left-inverse of i. □

Note 2.5.2. Suppose that the exact sequence (2.6) is right-split, i.e. there exists a group homomorphism $s : K \to G$ such that $p \circ s = \mathrm{id}_K$. Moreover, suppose that we have found such an s. Then we know that $G = i(H) \oplus s(K)$, and we can construct images of group homomorphism and direct sums of groups. The isomorphism $H \oplus K \xrightarrow{\sim} G$ is given by $(h, k) \mapsto i(h) + s(k)$, with inverse $G \xrightarrow{\sim} H \oplus K$ given by $g \mapsto (t(h), p(k))$, where t is a left splitting of i, that is, a group homomorphism such that $t \circ i = \mathrm{id}_H$.

2.6 Torsion subgroups, exponents, orders, cyclic decompositions

2.6.1 Computing the order of a finite abelian group

Recall that a finitely generated abelian group G represented by triple (m, n, f) is finite if and only if the basis of $\mathrm{im}\, f$ given by the algorithm in Theorem 2.1.6 has n elements.

Proposition 2.6.1. *There exists a deterministic polynomial-time algorithm that, given a finitely generated abelian group G, computes the order of G.*

Proof. To test if G is finite, we run the algorithm of Theorem 2.1.6 and check if the basis of $\mathrm{im}\, f$ has n elements. If so, we compute the determinant of the matrix whose columns are given by these n basis elements, which is then equal to $|G|$. Otherwise we conclude that G has infinite order. □

2.6.2 Computing the torsion subgroup of a finitely generated abelian group

Suppose we are given a finitely generated abelian group G. We have seen that we can determine if G is finite or not. If we find that it is not finite, we would like to find its torsion subgroup. To do this, we introduce a construction described in [68], at the end of Section 14.

Theorem 2.6.2. *There exists a deterministic polynomial-time algorithm that, given a finitely generated abelian group G, determines its torsion subgroup T, and produces an isomorphism $G \cong T \oplus G/T$.*

Proof. Suppose G is represented by triple (m, n, f). Let $L := \mathbb{Z}^n$ and $H = \mathrm{im}(f) \subseteq \mathbb{Z}^n$. Suppose $H \cong \mathbb{Z}^k$ and let $F \in \mathcal{M}_{n\times k}(\mathbb{Z})$ be a matrix whose columns are a $\mathbb{Z}$-basis of H. Consider the map

$$\phi : L \to \mathrm{Hom}(H, \mathbb{Z}), \quad \phi(x)(y) = \langle x, y\rangle,$$

for $x \in L$, $y \in H$ and $\langle\cdot,\cdot\rangle$ the standard inner product. Then ϕ is represented by the matrix $F^{\mathrm{t}} \in \mathcal{M}_{k\times n}(\mathbb{Z})$. This is because $\langle x, y\rangle = y^{\mathrm{t}}F^{\mathrm{t}}x$, for any $x \in L, y \in H$. We denote the kernel of this map by

$$H^{\perp} := \{x \in L \mid \langle x, H\rangle = 0\}.$$

Since $H^{\perp}$ is the kernel of a map between two free $\mathbb{Z}$-modules, Theorem 2.1.6 produces for us a basis $b_1 \ldots, b_n$ of L, where $b_1, \ldots, b_{n-k}$ is a $\mathbb{Z}$-basis of $H^{\perp}$.

We repeat the process above by considering the map

$$\phi' : L \to \mathrm{Hom}(H^{\perp}, \mathbb{Z}),$$

where the matrix representing ϕ' is given by the transpose of the matrix whose columns are $b_1, \ldots, b_{n-k}$. We denote the kernel of this map by

$$H^{\perp\perp} := \{a \in L \mid \langle a, H^{\perp}\rangle = 0\} = (\mathbb{Q} \cdot H) \cap L. \tag{2.7}$$

Thus,

$$L/H^{\perp\perp} \cong (L/H) \Big/ (L/H)_{\mathrm{tor}},$$

where $(L/H)_{\mathrm{tor}}$ denotes the torsion subgroup of L/H. Our goal becomes to split the exact sequence

$$0 \longrightarrow H^{\perp\perp}/H \longrightarrow L/H \longrightarrow L/H^{\perp\perp} \longrightarrow 0, \tag{2.8}$$

which we do using Proposition 2.5.1. □

2.6.3 Computing the order of a group element

Theorem 2.6.3. *There exists a deterministic polynomial-time algorithm that, given a finitely generated abelian group G and an element $g \in G$, determines the order of g.*

Proof. Consider the map $\psi : \mathbb{Z} \to G$, given by $1 \mapsto g$. If ψ is injective, which we can test, then g has infinite order. Otherwise, $\ker(\psi)$ is of the form $l\mathbb{Z}$, giving the order of g in G as equal to l. □

Note 2.6.4. This result depends heavily on the way we are representing G. Suppose $G = (\mathbb{Z}/n\mathbb{Z})^\times$, for some $n \in \mathbb{Z}_{>2}$. Suppose we choose to represent G by only giving the integer n. Then we can certainly carry out computations in G, but given the element $2 \in G$, we cannot in general efficiently compute its order without knowing the factorisation of n, and even then it is hard.

We can use this tool to decide if two elements are equal, by simply determining if their difference is equal to the zero element, i.e. if it has order 1.

Corollary 2.6.5. *There exists a deterministic polynomial-time algorithm that, given a finitely generated abelian group G and an element $g \in G$, determines if $g = 0_G$.*

Corollary 2.6.6. *There exists a deterministic polynomial-time algorithm that, given a finitely generated abelian group G and two elements $g, h \in G$, determines if $g = h$.*

2.6.4 Computing the exponent of a group

The exponent of a group G is computable as the generator over $\mathbb{Z}$ of

$$\ker\left(\mathbb{Z} \to \mathrm{Hom}(G, G),\ n \mapsto (x \mapsto nx)\right).$$

However, it is also useful to be able to exhibit an element of G of order equal to the exponent of G. To do this, we begin with a couple of preliminary results.

Lemma 2.6.7. *There exists a deterministic polynomial-time algorithm that, given a finite abelian group G and two elements $x, y \in G$ of orders n and m respectively, outputs an element z, expressed in terms of x and y, of order equal to* $\mathrm{lcm}(n, m)$.

Proof. Using Theorem 2.6.3, compute n and m, the orders of x and y, respectively. Apply the Coprime Base Algorithm (Theorem 1.1.2) to the set $\{n, m\}$ to obtain a set $\mathcal{P}$ of coprime divisors of nm and a factorisation $n = \prod_{p\in\mathcal{P}} p^{n_p}$ and $m = \prod_{p\in\mathcal{P}} p^{m_p}$, where $n_p, m_p \in \mathbb{Z}_{\geq 0}$. Define

$$n' = \prod_{\substack{p\in\mathcal{P}\\ n_p > m_p}} p^{n_p} \quad \text{and} \quad m' = \prod_{\substack{p\in\mathcal{P}\\ n_p \leq m_p}} p^{m_p}$$

Let $x' = \frac{n}{n'}x$ and $y' = \frac{m}{m'}y$. Since n' and m' are coprime, the order of $z := x' + y'$ is equal to $\mathrm{lcm}(n', m') = \mathrm{lcm}(n, m)$. □

Lemma 2.6.8. *Let $L = \mathbb{Z}^n$ and $H \subseteq L$ a subgroup. Suppose $B = \{b_1, \ldots, b_n\}$ is a basis of L. Then the exponent of L/H is equal to the lowest common multiple of the orders of $b_1, \ldots, b_n$ in L/H.*

Proof. Let $e_1, \ldots, e_n$ be the respective orders of $b_1 + H, \ldots, b_n + H$ in L/H. Let $l = \mathrm{lcm}_i(\{e_i\})$. By Lemma 2.6.7, there exists an element of L/H of order equal to l, so $\exp(L/H) \geq l$. Moreover, since B is a basis of L, every $x \in L/H$ is of the form $\sum_i \alpha_i b_i + H$, for some $\alpha_i \in \mathbb{Z}$. Then $lx \in H$, so x has order dividing l. Hence $\exp(L/H) = l$. □

This now enables us to compute the exponent of any given finite abelian group.

Theorem 2.6.9. *There exists a deterministic polynomial-time algorithm such that, given a finite abelian group G, computes the exponent of G and produces an element $g \in G$ of order equal to the exponent.*

Proof. We begin by determining the order of all basis vectors of L in L/H using Theorem 2.6.3. This gives a sequence of integers whose lowest common multiple is the exponent of L/H. Applying Proposition 2.6.7 repeatedly, we produce an element with the required property. □

2.6.5 Writing a finitely generated abelian group as a direct sum of cyclic groups

Let G be a finite abelian group of exponent n. Then we have seen that there exists $g \in G$ such that the order of g is n. Moreover, for any such g, the cyclic subgroup generated by it is a direct summand of G. This suggests a method of decomposing a finite abelian group into a direct sum of cyclic subgroups by computing the exponent of the group, producing an element of that order, quotienting out by the subgroup it generates and repeating the process for the remaining part.

Theorem 2.6.10. *There exists a deterministic polynomial-time algorithm that, given a finite abelian group G, produces a direct-sum-decomposition of G into cyclic subgroups.*

Proof. Suppose G is represented by triple (m, n, f), and put $L := \mathbb{Z}^n$ and $H := \mathrm{im}(f)$. Then $G = L/H$. By Theorem 2.6.9, we can compute e_1, the exponent of G, and produce an element $x_1 \in G$ of order e_1. The subgroup generated by x_1 is now a direct summand of G. We would now like to determine a subgroup $G_2 \leq G$ such that $G \cong G_2 \oplus \mathbb{Z}/e_1\mathbb{Z}$.

To do this, apply Proposition 2.5.1 to the exact sequence

$$0 \to \mathbb{Z}/e_1\mathbb{Z} \to G \to G/\mathbb{Z}x_1 \to 0.$$

Now replace G by G_2 and repeat. In the end we will have produced a positive integer $t \in \mathbb{Z}_{>0}$, a sequence of integers $e_1, \ldots, e_t \in \mathbb{Z}_{>0}$, a sequence of subgroups $G_1, \ldots, G_t \leq G$ such that the exponent of G_i is e_i, and a sequence of elements $x_1, \ldots, x_t$ such that $x_i \in G_i$ and the order of x_i is equal to e_i. Moreover, we have an isomorphism

$$G \xleftarrow[\sim]{\psi} \bigoplus_{i=1}^{t} \mathbb{Z}/e_i\mathbb{Z},$$

where $\psi : 1_{\mathbb{Z}/e_i\mathbb{Z}} \mapsto x_i$. The algorithm requires $t \leq \log_2 |G|$ iterations. □

Note 2.6.11. Another way to obtain a decomposition of G into cyclic subgroups is to apply Theorem 2.2.17 to the presentation matrix of G.

2.7 Homomorphism groups and tensor products reconsidered

We have already seen how to compute tensor products and homomorphism groups, without knowing a cyclic direct sum decomposition of the groups involved. The disadvantage of that approach is that the number of generators produced by the algorithm can get unnecessarily large. In this section we construct tensor products and homomorphism groups of finite abelian groups by making use of the fact that we can compute a cyclic direct sum decomposition.

Suppose

$$G_1 \cong \bigoplus_{i=1}^{t_1} \mathbb{Z}/c_i\mathbb{Z} \quad \text{and} \quad G_2 \cong \bigoplus_{j=1}^{t_2} \mathbb{Z}/d_i\mathbb{Z}, \tag{2.9}$$

for some $t_1, t_2 \in \mathbb{Z}_{>0}$ and $c_i, d_j \in \mathbb{Z}_{>0}$, for all $1 \leq i \leq t_1$ and $1 \leq j \leq t_2$.

Let $n, m \in \mathbb{Z}_{>0}$ and $d := \gcd(n, m)$. Then we have an isomorphism

$$\mathbb{Z}/n\mathbb{Z} \otimes \mathbb{Z}/m\mathbb{Z} \xrightarrow{\sim} \mathbb{Z}/d\mathbb{Z}, \quad x \otimes y \mapsto xy.$$

Similarly, every group homomorphism $\phi : \mathbb{Z}/n\mathbb{Z} \to \mathbb{Z}/m\mathbb{Z}$ is determined by the image of 1 in $\mathbb{Z}/m\mathbb{Z}$, which must be a multiple of $\frac{m}{d}$. This is because

$$n\phi(1) = \phi(n) \equiv 0 \mod m,$$

and so

$$\frac{n}{d}\phi(1) \equiv 0 \mod \frac{m}{d}.$$

Since n/d and m/d are coprime, it must be the case that $\phi(1) \equiv 0 \mod \frac{m}{d}$. Hence we have an isomorphism

$$\mathrm{Hom}(\mathbb{Z}/n\mathbb{Z}, \mathbb{Z}/m\mathbb{Z}) \xrightarrow{\sim} \mathbb{Z}/\gcd(n, m)\mathbb{Z}, \quad \phi \mapsto \left(\frac{m}{d}\right)^{-1} \phi(1).$$

Proposition 2.7.1. *There exists a deterministic polynomial-time algorithm that, given two finite abelian groups G_1, G_2 via direct-sum representations as in (2.9), computes*

(i) *the tensor product, $G_1 \otimes G_2$, together with the corresponding bilinear map $G_1 \times G_2 \to G_1 \otimes G_2$,*
(ii) *the homomorphism group, $\mathrm{Hom}(G_1, G_2)$, together with the corresponding bilinear map $\mathrm{Hom}(G_1, G_2) \times G_1 \to G_2$.*

Proof. Note that one can always obtain a direct-sum representation using Theorem

2.2.17. We then have that

$$\begin{aligned} G_1 \otimes G_2 &\cong \left(\bigoplus_{i=1}^{t_1} \mathbb{Z}/c_i\mathbb{Z}\right) \otimes \left(\bigoplus_{j=1}^{t_2} \mathbb{Z}/d_i\mathbb{Z}\right) \\ &\cong \bigoplus_{i,j=1}^{t_1,t_2} (\mathbb{Z}/c_i\mathbb{Z} \otimes \mathbb{Z}/d_j\mathbb{Z}) \\ &\cong \bigoplus_{i,j=1}^{t_1,t_2} \mathbb{Z}/\gcd(c_i, d_j)\mathbb{Z}, \end{aligned}$$

and all isomorphisms occurring are known and computable.

Similarly,

$$\begin{aligned} \mathrm{Hom}(G_1, G_2) &\cong \mathrm{Hom}\left(\bigoplus_{i=1}^{t_1} \mathbb{Z}/c_i\mathbb{Z}, \bigoplus_{j=1}^{t_2} \mathbb{Z}/d_i\mathbb{Z}\right) \\ &\cong \bigoplus_{i,j=1}^{t_1,t_2} \mathrm{Hom}(\mathbb{Z}/c_i\mathbb{Z}, \mathbb{Z}/d_j\mathbb{Z}) \\ &\cong \bigoplus_{i,j=1}^{t_1,t_2} \mathbb{Z}/\gcd(c_i, d_j)\mathbb{Z}, \end{aligned}$$

and all isomorphisms occurring are known and computable. □

2.8 Projective $\mathbb{Z}/m\mathbb{Z}$-modules

We have seen in Proposition 1.6.8, what the projective modules over $\mathbb{Z}/m\mathbb{Z}$ are, for $m \in \mathbb{Z}_{>0}$.

Suppose now we are given a finite abelian group A_1 and we would like to find the largest integer $m \mid \exp(A_1)$ such that A_1/mA_1 is projective as a module over $\mathbb{Z}/m\mathbb{Z}$.

Proposition 2.8.1. *There exists a deterministic polynomial-time algorithm that, given a finite abelian group A_1 computes the largest integer $m \in \mathbb{Z}_{>0}$ such that $m \mid \exp(A_1)$ and A_1/mA_1 is projective as a $\mathbb{Z}/m\mathbb{Z}$-module.*

Proof. Suppose

$$A_1 \cong \bigoplus_{\substack{p \in \mathcal{P} \\ a \in \mathbb{Z}_{>0}}} (\mathbb{Z}/p^a\mathbb{Z})^{n_{a,p}},$$

where $\mathcal{P}$ is a set of pairwise coprime integers greater than 1 and $n_{a,p} \in \mathbb{Z}_{\geq 0}$. Note that this is a situation we can reduce to if we first compute a decomposition of A_1 as a direct sum of cyclic subgroups using Theorem 2.6.10 and then apply the Coprime

Base Algorithm (Theorem 1.1.2) to the set of sizes of these cyclic components. For each $p \in \mathcal{P}$, set

$$\alpha_p = \begin{cases} \min\{a \mid n_{a,p} \neq 0\}, & \text{if } \exists a \text{ s. t. } n_{a,p} \neq 0 \\ 0, & \text{otherwise .} \end{cases}$$

Let

$$m = \prod_{p \in \mathcal{P}} p^{\alpha_p}.$$

Then

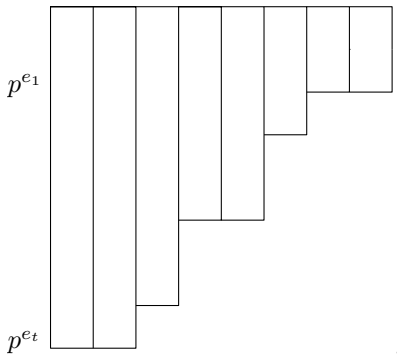
$$A_1/mA_1 \cong \bigoplus_{p \in \mathcal{P}} (\mathbb{Z}/p^{\alpha_p}\mathbb{Z})^{\sum_a n_{a,p}},$$

so is projective as a $\mathbb{Z}/m\mathbb{Z}$-module.

This value of m is the largest possible. To see this, it is enough to consider the case when $\mathcal{P} = \{p\}$ for some $p \in \mathbb{Z}_{>1}$. Suppose $m = p^{\beta_p}$, where β_p is a positive integer larger than α_p. Then A_1/mA_1 will have $\mathbb{Z}/p^{\alpha_p}\mathbb{Z}$ as a direct summand and thus cannot be projective as a $\mathbb{Z}/m\mathbb{Z}$-module.

□

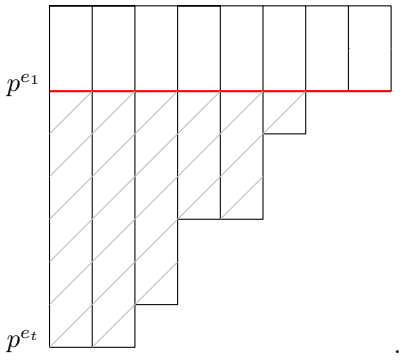
To see maximality of m graphically, let p be a prime and consider the p-group $A_1 = \bigoplus_{i=1}^{t} \mathbb{Z}/p^{e_i}\mathbb{Z}$, where $e_i \leq e_{i+1}$ and $e_1 > 0$. We represent A_1 by the following diagram:



. (2.10)

The number of vertical boxes is equal to the number of cyclic direct summands of A_1 and the height of each such box proportional to the length of the cyclic group it represents.

To make this into a free module, we need to "cut out a rectangle", so we need to cut along the smallest invariant, p^{e_1}, or along any other $p^{e'}$, for $e' \leq e_1$:



The remaining part is isomorphic to $(\mathbb{Z}/p^{e_1}\mathbb{Z})^t$ as an abelian group.

Suppose A_2 is a finite abelian group and we want to find the least integer $m' \mid \exp(A_2)$ such that $A_2/A_2[m']$ is projective over $\mathbb{Z}/\frac{\exp(A_2)}{m'}\mathbb{Z}$, where $A_2[m'] = \ker(A_2 \to A_2,\ x \mapsto m'x)$.

Proposition 2.8.2. *There exists a deterministic polynomial-time algorithm that, given a finite abelian group A_2 of exponent m, for some $m \in \mathbb{Z}_{>0}$, computes the smallest $m' \mid m$ such that $A_2/A_2[m']$ is projective over $\mathbb{Z}/\frac{m}{m'}\mathbb{Z}$.*

Proof. Suppose

$$A_2 \cong \bigoplus_{\substack{q \in \mathcal{Q} \\ b \in \mathbb{Z}_{>0}}} \left(\mathbb{Z}/q^b\mathbb{Z}\right)^{n_{b,q}}, \tag{2.11}$$

where $\mathcal{Q}$ is a set of coprime integers greater than 1 and $n_{b,q} \in \mathbb{Z}_{\geq 0}$. For each $q \in \mathcal{Q}$, let

$$\mu(q) = \begin{cases} \max\{b \mid n_{b,q} \neq 0\}, & \text{if } \exists b \text{ s.t. } n_{b,q} \neq 0 \\ 0, & \text{otherwise,} \end{cases}$$

and

$$\alpha(q) = \begin{cases} \max\{\{b \mid n_{b,q} \neq 0\} \backslash \mu(q)\}, & \text{if } \exists b \text{ s.t. } n_{b,q} \neq 0 \text{ and } b \neq \mu(q) \\ 0, & \text{otherwise.} \end{cases}$$

Then $\mu(q)$ is the largest power of q occurring as the exponent of one of the cyclic groups in the direct sum (2.11), and $\alpha(q)$ is the second largest power of q occurring. Set

$$m' = \prod_{q \in \mathcal{Q}} q^{\alpha(q)}.$$

Then

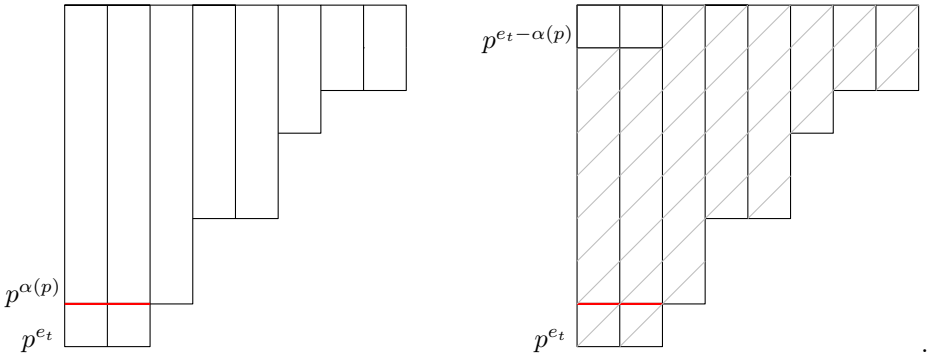
$$A_2/A_2[m'] = \bigoplus_{q \in \mathcal{Q}} \left(\mathbb{Z}/q^{\alpha(q)}\mathbb{Z}\right)^{n_{\mu(q),q}},$$

where $n_{0,q} := 0$.

This value of m' is the smallest possible. To see this, it is enough to consider the case when $\mathcal{Q} = \{p\}$, for some $p \in \mathbb{Z}_{>1}$. Suppose $m' = \beta(p)$, where $\beta(q)$ is a

positive integer smaller than $\alpha(q)$. Then $A_2/A_2[m]$ will have $\mathbb{Z}/p^{\alpha(q)-\beta(q)}\mathbb{Z}$ as a direct summand, and thus cannot be projective as a $\mathbb{Z}/\frac{m}{m'}\mathbb{Z}$-module. □

To see minimality of m' graphically, reduce again to the p-group case. Suppose $A_2 = \bigoplus_{i=1}^{t} \mathbb{Z}/p^{e_i}\mathbb{Z}$, where $e_i \leq e_{i+1}$ and $e_1 > 0$, and suppose that there are k copies of $\mathbb{Z}/p^{e_t}\mathbb{Z}$ in this sum. Then A_2 is represented by a diagram like in (2.10). This time however, to get a rectangle, we do not cut along the smallest invariant, but along the second largest one:



The remaining part is represented by the upper-left rectangle left unhatched, and is isomorphic to $(\mathbb{Z}/p^{e_t-\alpha(p)}\mathbb{Z})^k$.