MATHEMATICS OF COMPUTATION
Volume 65, Number 216
October 1996, Pages 1681–1699

HERMITE AND SMITH NORMAL FORM ALGORITHMS OVER DEDEKIND DOMAINS

HENRI COHEN

Abstract. We show how the usual algorithms valid over Euclidean domains, such as the Hermite Normal Form, the modular Hermite Normal Form and the Smith Normal Form can be extended to Dedekind rings. In a sequel to this paper, we will explain the use of these algorithms for computing in relative extensions of number fields.

The goal of this paper is to explain how to generalize to a Dedekind domain R many of the algorithms which are usually associated with a Euclidean domain, such as the Hermite Normal Form algorithm (including a modular version), and the Smith Normal Form algorithm. Since the goal of this paper is eminently practical, we will restrict our attention to the case where R is the ring of integers of a number field, for which we assume known a $\mathbb{Z}$-basis. Most of the algorithms can however be transposed to a more general context.

An immediate application of these algorithms (which was evidently our sole motivation) is to computing in relative extensions of number fields. This can now indeed be done very easily, as we will show in a subsequent paper ([3]).

These ideas have already been used by Bosma and Pohst [1].

Notations: R will always denote the ring of integers $\mathbb{Z}_K$ of a number field K (although most of the results apply to general Dedekind domains), and K is the field of fractions of R. Unless otherwise specified, an *ideal* of R will always mean a nonzero fractional ideal.

1. Basic algorithms

We start by some preliminary but essential algorithms.

Proposition 1.1. *Given two coprime integral ideals $\mathfrak{a}$ and $\mathfrak{b}$ in R, we can find in polynomial time elements $a \in \mathfrak{a}$ and $b \in \mathfrak{b}$ such that $a + b = 1$.*

Proof. We can assume that the ideals are given by their Hermite Normal Form (HNF) matrices A and B on some $\mathbb{Z}$-basis of R whose first element is always assumed to be equal to 1; otherwise it is easy to reduce to this case. Call C the $n \times 2n$ matrix obtained by concatenating A and B. Using one of the polynomial-time algorithms for HNF reduction (see, for example, [5]), we can find a $2n \times 2n$ unimodular matrix U such that CU is the concatenation of the $n \times n$ zero matrix and the $n \times n$ identity matrix, since by assumption $\mathfrak{a}$ and $\mathfrak{b}$ are coprime. It follows that if Z is the $(n+1)$st

Received by the editor January 11, 1995 and, in revised form, July 19, 1995.
1991 *Mathematics Subject Classification.* Primary 11Y40.
Key words and phrases. Dedekind domain, Hermite normal form, Smith normal form, relative extensions of number fields.

column of U, then $CZ = [1, 0, \ldots, 0]$ (note that if we use the algorithm of [6], we will find a permutation matrix instead of the identity matrix and in that case it is not the $(n+1)$st column but some other column of U which must be used). If we split Z into its upper half X and its lower half Y, it is clear that AX represents an element $a \in \mathfrak{a}$ and BY respresents an element $b \in \mathfrak{b}$ such that $a + b = 1$. □

Implementation Remarks.

(1) It was of course not really necessary in the proof that the ideals be given by HNF matrices, but only by $\mathbb{Z}$-bases. However, if we do really have HNF bases, the first column of the matrix A of $\mathfrak{a}$ will be a generator z_a of $\mathfrak{a} \cap \mathbb{Z}$, and similarly the first column of B will be a generator z_b of $\mathfrak{b} \cap \mathbb{Z}$. Now it frequently will happen that z_a and z_b are coprime. In this case, the usual extended Euclidean algorithm will easily find u and v such that $uz_a + vz_b = 1$, and we can take $a = uz_a$ and $b = vz_b$.

(2) Since the algorithm underlying this proposition will be absolutely basic to all our algorithms on Dedekind domains, we must insure that it will give results which are as reasonable as possible. Indeed, the elements a and b are of course not unique, and can be modified by adding and subtracting from a and b respectively some element of the ideal product $\mathfrak{ab}$. Hence it would be nice to have an element r such that $a - r \in \mathfrak{ab}$ and r is "small" (and then we replace a by $a - r$ and b by $b + r = 1 - (a - r)$ which will also be "small". We will see below (Algorithm 2.12) how this can be done reasonably well.

(3) This is the most important place of this paper where we use specifically the fact that the Dedekind domain R is the ring of integers of a number field, so as to be able to compute a and b in polynomial time.

We now come to a theorem which is trivial to prove, but is the basic tool for our algorithms. It is a generalization to Dedekind domains of the extended Euclidean algorithm, as follows.

Theorem 1.2. *Let $\mathfrak{a}$ and $\mathfrak{b}$ two (fractional) ideals in R, let a and b be two elements of K not both equal to zero, and set $\mathfrak{d} = a\mathfrak{a} + b\mathfrak{b}$. There exists $u \in \mathfrak{ad}^{-1}$ and $v \in \mathfrak{bd}^{-1}$ such that $au + bv = 1$, and these elements can be found in polynomial time.*

Proof. If a (resp. b) is equal to zero we can take $(u, v) = (0, 1/b)$ (resp. $(u, v) = (1/a, 0)$) since in that case we have $1/b \in \mathfrak{bd}^{-1} = R/b$ (resp. $1/a \in \mathfrak{ad}^{-1} = R/a$). So assume a and b are nonzero.

Set $I = a\mathfrak{a}\mathfrak{d}^{-1}$ and $J = b\mathfrak{b}\mathfrak{d}^{-1}$. By the definition of $\mathfrak{d}^{-1}$, I and J are integral ideals and we have $I + J = R$. By Proposition 1.1 we can thus find in polynomial time $e \in I$ and $f \in J$ such that $e + f = 1$, and clearly $u = e/a$ and $v = f/b$ satisfy the conditions of the lemma. □

Remark. Although this proposition is very simple, we will see that the essential conditions that $u \in \mathfrak{ad}^{-1}$ and $v \in \mathfrak{bd}^{-1}$ bring as much rigidity into the problem as in the case of Euclidean domains, and this proposition will be constantly used instead of the extended Euclidean algorithm. It is in fact clear that it *is* an exact generalization of the extended Euclidean algorithm. Note that even when R is a principal ideal domain, this lemma is useful, since R is not necessarily Euclidean.

We also need the following.

Proposition 1.3. *Let $\mathfrak{a}$, $\mathfrak{b}$, $\mathfrak{c}$, $\mathfrak{d}$ be fractional ideals of R, and let a, b, c, d be elements of K. Set $e = ad - bc$, and assume that*

$$\mathfrak{ab} = e\mathfrak{cd}, \quad a \in \mathfrak{ac}^{-1}, \quad b \in \mathfrak{bc}^{-1}, \quad c \in \mathfrak{ad}^{-1}, \quad d \in \mathfrak{bd}^{-1} \ .$$

Finally, let x and y be two elements of an R-module M, and set

$$\begin{pmatrix} x' & y' \end{pmatrix} = \begin{pmatrix} x & y \end{pmatrix} \begin{pmatrix} a & c \\ b & d \end{pmatrix} \ .$$

Then

$$\mathfrak{a}x + \mathfrak{b}y = \mathfrak{c}x' + \mathfrak{d}y' \ .$$

Proof. We have $x' = ax + by$ and $y' = cx + dy$, hence

$$\mathfrak{c}x' + \mathfrak{d}y' \subset (a\mathfrak{c} + c\mathfrak{d})x + (b\mathfrak{c} + d\mathfrak{d})y \subset \mathfrak{a}x + \mathfrak{b}y \ .$$

Conversely, we have $x = (dx' - by')/e$ and $y = (-cx' + ay')/e$, hence

$$\mathfrak{a}x + \mathfrak{b}y \subset \frac{1}{e}(\mathfrak{ab}\mathfrak{d}^{-1}x' + \mathfrak{ab}\mathfrak{c}^{-1}y') \ ,$$

and since $\mathfrak{ab} \subset e\mathfrak{cd}$,

$$\mathfrak{a}x + \mathfrak{b}y \subset \mathfrak{cd}(\mathfrak{d}^{-1}x' + \mathfrak{c}^{-1}y') = \mathfrak{c}x' + \mathfrak{d}y' \ ,$$

thus showing the double inclusion.

Note that although we have used only the inclusion $\mathfrak{ab} \subset e\mathfrak{cd}$ in the proof, the hypotheses on a, b, c and d imply that $e\mathfrak{cd} \subset \mathfrak{ab}$, so we must have equality. □

Corollary 1.4. *Let $\mathfrak{a}$, $\mathfrak{b}$ be two ideals, a and b two elements of K not both zero, $\mathfrak{d} = a\mathfrak{a} + b\mathfrak{b}$ and $u \in \mathfrak{ad}^{-1}$, $v \in \mathfrak{bd}^{-1}$ such that $au + bv = 1$ as given by Theorem* 1.2.

Let x and y be two elements of an R-module M, and set

$$\begin{pmatrix} x' & y' \end{pmatrix} = \begin{pmatrix} x & y \end{pmatrix} \begin{pmatrix} b & u \\ -a & v \end{pmatrix} \ .$$

Then

$$\mathfrak{a}x + \mathfrak{b}y = \mathfrak{ab}\mathfrak{d}^{-1}x' + \mathfrak{d}y' \ .$$

Proof. Since $b \in \mathfrak{b}^{-1}\mathfrak{d}$ and $a \in \mathfrak{a}^{-1}\mathfrak{d}$, this is clearly a special case of Proposition 1.3. □

Corollary 1.5. *Let $\mathfrak{a}$, $\mathfrak{b}$ be two ideals. Assume that a, b, c and d are four elements of K such that*

$$ad - bc = 1, \quad a \in \mathfrak{a}, \quad b \in \mathfrak{b}, \quad c \in \mathfrak{b}^{-1}, \quad d \in \mathfrak{a}^{-1} \ .$$

Let x and y be two elements of an R-module M, and set

$$\begin{pmatrix} x' & y' \end{pmatrix} = \begin{pmatrix} x & y \end{pmatrix} \begin{pmatrix} a & c \\ b & d \end{pmatrix} \ .$$

Then

$$\mathfrak{a}x + \mathfrak{b}y = Rx' + \mathfrak{ab}y' \ .$$

Proof. This is also trivially a special case of Proposition 1.3. We will see below (Proposition 1.11) how to find a, b, c, d, given $\mathfrak{a}$ and $\mathfrak{b}$. □

Remarks.

(1) The type of elementary transformation described in Proposition 1.3, and in particular in its two corollaries above, will be the only one that we are allowed to use. For example, if we want simply to replace x by $x - qy$ for some q in the field K (which is the usual elementary transformation), we must have $q \in \mathfrak{b}\mathfrak{a}^{-1}$, as can easily be checked.

(2) With the notations of Proposition 1.3, note that we also have the formal equality

$$[\mathfrak{c}^{-1}, \mathfrak{d}^{-1}] = [\mathfrak{a}^{-1}, \mathfrak{b}^{-1}] \begin{pmatrix} a & c \\ b & d \end{pmatrix} \ .$$

Indeed, since $a \in \mathfrak{a}\mathfrak{c}^{-1}$ and $b \in \mathfrak{b}\mathfrak{c}^{-1}$ it is clear that $a\mathfrak{a}^{-1} + b\mathfrak{b}^{-1} \subset \mathfrak{c}^{-1}$. Conversely, since $e = ad - bc$ we have $e \in a\mathfrak{b}\mathfrak{d}^{-1} + b\mathfrak{a}\mathfrak{d}^{-1}$, hence $e\mathfrak{c}\mathfrak{d} \subset a\mathfrak{b}\mathfrak{c} + b\mathfrak{a}\mathfrak{c}$, and since $\mathfrak{a}\mathfrak{b} = e\mathfrak{c}\mathfrak{d}$ we obtain the reverse inclusion $\mathfrak{c}^{-1} \subset a\mathfrak{a}^{-1} + b\mathfrak{b}^{-1}$. The second equality $\mathfrak{d}^{-1} = c\mathfrak{a}^{-1} + d\mathfrak{b}^{-1}$ is proved in a similar manner.

As we will see in §4 below, the "real" reason for these identities is that $\mathfrak{a}^{-1}$, for every ideal $\mathfrak{a}$, can be canonically identified with the set of R-linear maps from $\mathfrak{a}$ to R.

It will also be useful (although not essential) to have some algorithms linked to the approximation theorem in Dedekind domains. We give straightforward deterministic versions, but in practice it is much better to use other methods.

Proposition 1.6. *Given ideals $\mathfrak{a}_i$ for $1 \le i \le k$ whose sum is equal to R, we can find in polynomial time elements $a_i \in \mathfrak{a}_i$ such that $\sum_i a_i = 1$.*

Proof. Same proof as for Proposition 1.1, except that we concatenate the k HNF matrices of the ideals and that we split Z into k pieces at the end. Note that the matrix U will be an $nk \times nk$ unimodular matrix, and this can become quite large. □

Proposition 1.7. *Let S be a finite set of prime ideals of R and let $(e_{\mathfrak{p}})_{\mathfrak{p}\in S} \in \mathbb{Z}^S$. Then there exists a polynomial-time algorithm which finds $a \in K$ such that $v_{\mathfrak{p}}(a) = e_{\mathfrak{p}}$ for $\mathfrak{p} \in S$ and $v_{\mathfrak{p}}(a) \ge 0$ for $\mathfrak{p} \notin S$.*

Proof. We can write $e_{\mathfrak{p}} = f_{\mathfrak{p}} - g_{\mathfrak{p}}$ for $f_{\mathfrak{p}} \ge 0$ and $g_{\mathfrak{p}} \ge 0$. If we can find n (resp. d) such that the conditions are satisfied with $e_{\mathfrak{p}}$ replaced by $f_{\mathfrak{p}}$ (resp. $g_{\mathfrak{p}}$), it is clear that $a = n/d$ satisfies our conditions. Thus, we may assume that $e_{\mathfrak{p}} \ge 0$ for $\mathfrak{p} \in S$. Following the classical proof (see, for example, [2]), we compute the ideal product

$$I = \prod_{\mathfrak{p}\in S} \mathfrak{p}^{e_{\mathfrak{p}}+1}$$

and we set for each $\mathfrak{p} \in S$

$$\mathfrak{a}_{\mathfrak{p}} = I \cdot \mathfrak{p}^{-e_{\mathfrak{p}}-1} \ .$$

Then the $\mathfrak{a}_{\mathfrak{p}}$ are integral ideals which sum to R, so by Proposition 1.7 we can find in polynomial time $a_{\mathfrak{p}} \in \mathfrak{a}_{\mathfrak{p}}$ whose sum is equal to 1. Furthermore, we can find $b_{\mathfrak{p}} \in \mathfrak{p}^{e_{\mathfrak{p}}} \setminus \mathfrak{p}^{e_{\mathfrak{p}}+1}$ (for example by taking the $e_{\mathfrak{p}}$th power of an element of $\mathfrak{p} \setminus \mathfrak{p}^2$ which can be found in polynomial time). Then $a = \sum_{\mathfrak{p}\in S} a_{\mathfrak{p}} b_{\mathfrak{p}}$ is a solution to our problem. □

Corollary 1.8. *Given two integral ideals $\mathfrak{a}$ and $\mathfrak{b}$ of R such that the factorization of the norm of $\mathfrak{b}$ is known, there exists a polynomial-time algorithm which finds $x \in K$ such that $x\mathfrak{a}$ is an integral ideal coprime to $\mathfrak{b}$, and similarly finds $y \in K$ such that $y\mathfrak{a}^{-1}$ is an integral ideal coprime to $\mathfrak{b}$.*

Proof. For x, apply Proposition 1.7 to S equal to the prime ideal factors of $\mathfrak{b}$ and to $e_{\mathfrak{p}} = -v_{\mathfrak{p}}(\mathfrak{a})$ for all $\mathfrak{p} \in S$. For y, apply Proposition 1.7 to S equal to the prime ideal factors of $\mathfrak{a}$ and $\mathfrak{b}$ and to $e_{\mathfrak{p}} = v_{\mathfrak{p}}(\mathfrak{a})$ for all $\mathfrak{p} \in S$. □

Proposition 1.9. *Let $\mathfrak{a}$ be an integral ideal of R and $a \in \mathfrak{a}$, $a \neq 0$. Assume that the prime ideal factorization of a is known. Then there exists a polynomial-time algorithm which finds $b \in \mathfrak{a}$ such that $\mathfrak{a} = aR + bR$.*

Proof. Write $aR = \prod_{\mathfrak{p}} \mathfrak{p}^{e_{\mathfrak{p}}}$ with $e_{\mathfrak{p}} \geq 0$. Thus, we have $\mathfrak{a} = \prod_{\mathfrak{p}} \mathfrak{p}^{v_{\mathfrak{p}}(\mathfrak{a})}$ with $0 \leq v_{\mathfrak{p}}(\mathfrak{a}) \leq e_{\mathfrak{p}}$. By Proposition 1.7 we can in polynomial time find $b \in R$ such that $v_{\mathfrak{p}}(b) = v_{\mathfrak{p}}(\mathfrak{a})$ for all $\mathfrak{p} \mid a$, and by looking at $\mathfrak{p}$-adic valuations, it is clear that $\mathfrak{a} = aR + bR$. □

Remarks. Recall that R is the ring of integers of a number field. Then

(1) If $\mathfrak{p}$ is a prime ideal given by a $\mathbb{Z}$-basis, the above proposition shows that we can find in polynomial time a two-element generating system for $\mathfrak{p}$. Indeed, we take $a = p$, and using the polynomial-time algorithm of Buchmann and Lenstra (see [2]), we can factor pR into prime ideals so the condition is satisfied.
(2) To factor a it is enough to factor the absolute norm $\mathcal{N}(a) \in \mathbb{Z}$ of a since we can use the Buchmann-Lenstra algorithm for factoring into prime ideals the prime factors of $\mathcal{N}(a)$, and use the algorithm explained in [2] for computing $\mathfrak{p}$-adic valuations, which is also polynomial-time as soon as a two-element generating set is known for every prime ideal $\mathfrak{p}$, which is the case by (1).
(3) As mentioned earlier, it is much faster in practice to perform a search for the elements that we need in Corollary 1.8 and Proposition 1.9. Of course, the time to perform this search is a priori exponential, but in practice it will always be very fast.

The strong form of the approximation theorem can be dealt with in the same manner:

Proposition 1.10. *Let S be a finite set of prime ideals of R, let $(e_{\mathfrak{p}})_{\mathfrak{p} \in S} \in \mathbb{Z}^S$, and let $(x_{\mathfrak{p}})_{\mathfrak{p} \in S} \in K^S$. Then there exists a polynomial-time algorithm which finds $x \in K$ such that $v_{\mathfrak{p}}(x - x_{\mathfrak{p}}) = e_{\mathfrak{p}}$ for $\mathfrak{p} \in S$ and $v_{\mathfrak{p}}(x) \geq 0$ for $\mathfrak{p} \notin S$.*

Proof. Assume first that the $e_{\mathfrak{p}}$ are nonnegative and $x_{\mathfrak{p}} \in R$. Then we introduce the same ideals I and $\mathfrak{a}_{\mathfrak{p}}$, and elements $a_{\mathfrak{p}}$ as in the proof of Proposition 1.7. If we set

$$x = \sum_{\mathfrak{p} \in S} a_{\mathfrak{p}} x_{\mathfrak{p}} \ ,$$

it is easy to see that x satisfies the required conditions.

Consider now the general case. Let $d \in R$ be a common denominator for the $x_{\mathfrak{p}}$, and multiply d by suitable elements of R so that $e_{\mathfrak{p}} + v_{\mathfrak{p}}(d) \geq 0$ for all $\mathfrak{p} \in S$. According to what we have just proved, there exists a $y \in R$ such that

$$\forall \mathfrak{p} \in S, \ v_{\mathfrak{p}}(y - dx_{\mathfrak{p}}) = e_{\mathfrak{p}} + v_{\mathfrak{p}}(d)$$

and

$$\forall \mathfrak{p} \mid d,\ \mathfrak{p} \notin S,\ v_{\mathfrak{p}}(y - dx_{\mathfrak{p}}) = v_{\mathfrak{p}}(d)\ .$$

It follows that $x = y/d$ satisfies the given conditions. □

Finally, we show how elements satisfying Corollary 1.5 can be found.

Proposition 1.11. *Let $\mathfrak{a}$ and $\mathfrak{b}$ be two (fractional) ideals in R. Assume that the prime ideal factorization of $\mathfrak{a}$ or of $\mathfrak{b}$ is known. Then it is possible to find in polynomial time elements $a \in \mathfrak{a}$, $b \in \mathfrak{b}$, $c \in \mathfrak{b}^{-1}$ and $d \in \mathfrak{a}^{-1}$ such that $ad - bc = 1$.*

Proof. Multiplying $\mathfrak{a}$ and $\mathfrak{b}$ by an element of $\mathbb{Q}^*$, if necessary, we can reduce to the case where $\mathfrak{a}$ and $\mathfrak{b}$ are integral ideals. Assume for example that the factorization of $\mathfrak{b}$ is known. According to Corollary 1.8 we can find in polynomial time $a \in R$ such that $a\mathfrak{a}^{-1}$ is an integral ideal (i.e. $a \in \mathfrak{a}$) and coprime to $\mathfrak{b}$. According to Proposition 1.1 we thus can find $e \in a\mathfrak{a}^{-1}$ and $f \in \mathfrak{b}$ such that $e + f = 1$. Clearly $b = f$, $c = -1$, $d = e/a$ satisfy the required conditions. □

Remark. It is an interesting question whether this can be done in polynomial time without knowing the prime ideal factorization of either $\mathfrak{a}$ or $\mathfrak{b}$ (or equivalently of the norm of either $\mathfrak{a}$ or $\mathfrak{b}$, since we work in number fields). H. W. Lenstra informs me that this can indeed be done using *factor refinement*, which we will not explain here.

2. The Hermite Normal Form algorithm over Dedekind domains

We recall the following theorem, which summarizes the main properties of finitely generated modules over Dedekind domains.

Theorem 2.1. *Let R be a Dedekind domain, and let M be a finitely generated R-module.*

(1) *If*

$$M_{\mathrm{tors}} = \{x \in M/\ \exists a \in R, ax = 0\}$$

is the torsion submodule of M, there exists a torsion-free submodule N of M such that

$$M = M_{\mathrm{tors}} \oplus N \qquad \text{and} \qquad N \simeq M/M_{\mathrm{tors}}\ .$$

(2) *A finitely generated R-module M is torsion free if and only if M is a projective module. If $V = MK$ is the vector space spanned by M, and if $n = \dim_K(V)$ is the rank of M, there exist (fractional) ideals $\mathfrak{a}_i$ and elements $\omega_i \in V$ such that*

$$M = \mathfrak{a}_1\omega_1 \oplus \mathfrak{a}_2\omega_2 \oplus \cdots \oplus \mathfrak{a}_n\omega_n\ .$$

The class of the product $\mathfrak{a} = \mathfrak{a}_1\mathfrak{a}_2\cdots\mathfrak{a}_n$ in the class group of R depends only on the module M and is called the Steinitz class *of M. The module M is a free R-module if and only if its Steinitz class is equal to the trivial class.*

(3) *Let M be a finitely generated torsion module. Then there exist integral ideals $\mathfrak{d}_i$ of R and elements $\omega_i \in M$ such that*

$$M = (R/\mathfrak{d}_1)\omega_1 \oplus \cdots \oplus (R/\mathfrak{d}_n)\omega_n$$

and $\mathfrak{d}_{i-1} \subset \mathfrak{d}_i$ for $2 \le i \le n$.

In this section we will only consider finitely generated torsion-free R-modules and refer to §4 for torsion modules. In view of the above theorem, it is natural to give the following definition.

Definition 2.2. Let M be a finitely generated torsion-free R-module, and set $V = MK$. If $\mathfrak{a}_i$ are (fractional) ideals of R and ω_i are elements of V, we say that $(\omega_i, \mathfrak{a}_i)_{1\le i\le k}$ form a *pseudogenerating set* for M if

$$M = \mathfrak{a}_1\omega_1 + \cdots + \mathfrak{a}_k\omega_k \ .$$

We say that it is a *pseudobasis* of M if the sum is direct, i.e., if

$$M = \mathfrak{a}_1\omega_1 \oplus \cdots \oplus \mathfrak{a}_k\omega_k \ .$$

Note that according to Theorem 2.1, any finitely generated torsion-free module has a pseudobasis.

Let $(\omega_i, \mathfrak{a}_i)_{1\le i\le n}$ be a pseudobasis of M. Then n is equal to the rank of M. It is clear that, among other transformations, we can multiply $\mathfrak{a}_i$ by a nonzero element of K as long as we divide ω_i by the same element, and we will still have a pseudobasis. In particular, if so desired, we may assume that the $\mathfrak{a}_i$ are integral ideals, or that the ω_i are elements of M. On the other hand, it is in general not possible to have both properties at once. A simple example is when $M = \mathfrak{a}$ a nonprincipal primitive integral ideal. Then the general pseudobasis of M is $(a, \mathfrak{a}/a)$, and so to have both an element of M and an integral ideal, we would need $a \in \mathfrak{a}$ and $\mathfrak{a}/a \subset R$, which is equivalent to $\mathfrak{a} = aR$, contrary to our choice of $\mathfrak{a}$.

Furthermore, restricting either to elements of M or to integral ideals would be too rigid for algorithmic purposes, so it is preferable not to choose a pseudobasis of a particular type for the moment. We will systematically represent finitely generated torsion-free R-modules by pseudobases. To be able to do this, we need to know how to compute such pseudobases, and how to perform usual operations on these pseudobases. As for the case $R = \mathbb{Z}$, the basic algorithm for doing this is the Hermite Normal Form algorithm, and we will see that such an algorithm does indeed exist. Before doing this, however, let us see how one can go from one basis to another.

The following proposition is a generalization of Proposition 1.3.

Proposition 2.3. *Let $(\omega_j, \mathfrak{a}_j)$ and $(\eta_j, \mathfrak{b}_j)$ be two pseudobases for an R-module M, and let $U = (u_{i,j})$ be the $n \times n$ matrix giving the η_j in terms of the ω_j (so that $[\eta_1, \ldots, \eta_n] = [\omega_1, \ldots, \omega_n]U$).*

Set $\mathfrak{a} = \mathfrak{a}_1 \cdots \mathfrak{a}_n$ and $\mathfrak{b} = \mathfrak{b}_1 \cdots \mathfrak{b}_n$. Then $u_{i,j} \in \mathfrak{a}_i\mathfrak{b}_j^{-1}$ and $\mathfrak{a} = \det(U)\mathfrak{b}$ (note that by Theorem 2.1, we know that $\mathfrak{a}$ and $\mathfrak{b}$ are in the same ideal class). Conversely, if there exist ideals $\mathfrak{b}_j$ such that $\mathfrak{a} = \det(U)\mathfrak{b}$ (with $\mathfrak{b} = \mathfrak{b}_1 \cdots \mathfrak{b}_n$) and $u_{i,j} \in \mathfrak{a}_i\mathfrak{b}_j^{-1}$, then $(\eta_j, \mathfrak{b}_j)$ is a pseudobasis of M, where the η_j are given in terms of the ω_j by the columns of U.

Proof. Since

$$\eta_j \in \mathfrak{b}_j^{-1}M = \mathfrak{b}_j^{-1}\bigoplus_{i=1}^{n}\mathfrak{a}_i\omega_i = \bigoplus_{i=1}^{n}\mathfrak{a}_i\mathfrak{b}_j^{-1}\omega_i \ ,$$

it follows that $u_{i,j} \in \mathfrak{a}_i\mathfrak{b}_j^{-1}$.

Now it is easily proven by linearity or by induction on n that $e = \det(U) \in \mathfrak{a}\mathfrak{b}^{-1}$, so $e\mathfrak{b} \subset \mathfrak{a}$. Similarly, if V is the inverse matrix of U which expresses the ω_j in terms

of the η_j, then $\det(V) \in \mathfrak{b}\mathfrak{a}^{-1}$. But since $\det(U)\det(V) = 1$ we have $\det(V) = 1/e$, hence $\mathfrak{a}/e \subset \mathfrak{b}$, i.e., $\mathfrak{a} \subset e\mathfrak{b}$, from which it follows that $\mathfrak{a} = e\mathfrak{b}$.

Conversely, if U has the properties above, then by looking at the adjoint matrix of U, it easily follows that its inverse V is of a similar form with $\mathfrak{a}$ and $\mathfrak{b}$ exchanged (it is of course essential that $\mathfrak{a} = \det(U)\mathfrak{b}$). If $X = [x_1, \dots, x_n]^t$ is the column vector of components of an element m of M in the pseudobasis $(\omega_j, \mathfrak{a}_j)$, then $m = [\omega_1, \dots, \omega_n]X = [\eta_1, \dots, \eta_n]VX$, and $VX = [y_1, \dots, y_n]^t$ satisfies $y_i \in \mathfrak{b}_i$ for $1 \le i \le n$. Since the y_i are unique, this shows that $(\eta_j, \mathfrak{b}_j)$ is a pseudobasis of M, thus proving the proposition. □

It is clear that Proposition 1.3 is the special case $n = 2$ of this proposition. However, since that special case is going to be used constantly, we presented it separately.

Corollary 2.4. *Let M be a finitely generated torsion-free R-module together with a nondegenerate bilinear pairing $T(x, y)$ from $M \times M$ to R (for example $M = \mathbb{Z}_L$, where L is a number field containing K, and $T(x, y) = \mathrm{Tr}_{L/K}(x \cdot y)$). For any pseudobasis $\mathcal{B} = (\omega_j, \mathfrak{a}_j)$ of M, let $\det(\mathcal{B})$ be the ideal defined by $\det(\mathcal{B}) = \det(T(\omega_i, \omega_j))\mathfrak{a}^2$, where as usual $\mathfrak{a} = \mathfrak{a}_1 \cdots \mathfrak{a}_n$. Then if $\mathcal{B}' = (\eta_j, \mathfrak{b}_j)$ is another pseudobasis of M, we have $\det(\mathcal{B}') = \det(\mathcal{B})$.*

Proof. Note that since $\omega_j \notin M$ in general, in the definition above we extend the bilinear form T to $V \times V$ (where $V = MK$) by bilinearity. Let U be the matrix expressing the η_j in terms of the ω_j. We know that $\mathfrak{a} = \det(U)\mathfrak{b}$. By bilinearity, it is clear that if G (resp G') is the matrix of the $T(\omega_i, \omega_j)$ (resp. $T(\eta_i, \eta_j)$), then $G' = U^t G U$. It follows that

$$\det(\mathcal{B}') = \det(G')\mathfrak{b}^2 = \det(G)\det(U)^2\mathfrak{a}^2/\det(U)^2 = \det(G)\mathfrak{a}^2 = \det(\mathcal{B}). \quad \square$$

Since $\det(\mathcal{B})$ does not depend on the chosen pseudobasis $\mathcal{B}$, we will denote it by $\mathrm{disc}(M)$ and call it the *discriminant ideal* of M.

Remark. We can also define $\det(T(\omega_i, \omega_j))$ as an element $d(M) \in K^*/K^{*2}$, since under a change of pseudobasis this determinant is multiplied by $\det(U)^2 \in K^{*2}$. The pair $(\mathrm{disc}(M), d(M))$ will simply be called the *discriminant* of M. Note that knowledge of one of the components of the pair does not imply knowledge of the other, hence the pair itself is useful. In the absolute case where $M = \mathbb{Z}_K$ is the ring of integers of a number field K, considered as a $\mathbb{Z}$-module, the discriminant ideal $\mathrm{disc}(M)$ gives the absolute value of the usual discriminant, and $d(M)$ gives its sign (and some other information already contained in $\mathrm{disc}(M)$).

The main theorem of this section is that the notion of Hermite Normal Form can be extended to Dedekind domains. As is well known, the Hermite Normal Form algorithm is a direct generalization of the extended Euclidean algorithm. Since we now have such an algorithm available to us (Corollary 1.4), it is not surprising that this can be done.

We first introduce a definition. Let $A = (a_{i,j})$ be an $n \times k$ matrix with coefficients in K, and $I = (\mathfrak{a}_i)$ a list of k fractional ideals. We will call the data (A, I) a *pseudomatrix*, and the module associated with this pseudomatrix will be the module $M = \sum_{1 \le j \le k} \mathfrak{a}_j A_j$, where the A_j are the columns of A, so that $(A_j, \mathfrak{a}_j)$ is a pseudogenerating set for M.

Theorem 2.5 (Hermite Normal Form in Dedekind domains)**.** *Let (A, I) be a pseudomatrix, where $I = (\mathfrak{a}_i)$ is a list of k fractional ideals, and $A = (a_{i,j})$ is an $n \times k$ matrix. Assume that A is of rank n (so $k \geq n$) with coefficients in the field of fractions K of R (we could just as easily consider the case of a matrix of lower rank). Let $M = \sum_j \mathfrak{a}_j A_j$ be the R-module associated with the pseudomatrix (A, I). Then there exist k nonzero ideals $(\mathfrak{b}_j)_{1 \leq j \leq k}$ and a $k \times k$ matrix $U = (u_{i,j})$ satisfying the following conditions. Set $\mathfrak{a} = \mathfrak{a}_1 \cdots \mathfrak{a}_k$, $\mathfrak{b} = \mathfrak{b}_1 \cdots \mathfrak{b}_k$. Then*

(1) $\mathfrak{a} = \det(U)\mathfrak{b}$.

(2) *The matrix AU is of the following form:*

$$AU = \begin{pmatrix} 0 & 0 & \dots & 0 & 1 & * & \dots & * \\ 0 & 0 & \dots & 0 & 0 & 1 & \dots & * \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 0 & \dots & 0 & 1 \end{pmatrix},$$

where the first $k - n$ columns are zero.

(3) *If we call ω_j the elements corresponding to the nonzero columns of AU and $\mathfrak{c}_j = \mathfrak{b}_{k-n+j}$ for $1 \leq j \leq n$, then*

$$M = \mathfrak{c}_1\omega_1 \oplus \cdots \oplus \mathfrak{c}_n\omega_n \ ;$$

in other words, $(\omega_j, \mathfrak{c}_j)$ is a pseudobasis of M.

Proof. We give the proof as an algorithm, very similar to Algorithm 2.4.5 of [2] (which is the naïve HNF algorithm).

Algorithm 2.6 (HNF algorithm in Dedekind domains)**.** Given an $n \times k$ matrix $A = (a_{i,j})$ of rank n, and k (fractional) ideals $\mathfrak{a}_j$ in a number field K, this algorithm computes k ideals $\mathfrak{b}_j$ and a $k \times k$ matrix U such that this data satisfies the conditions of Theorem 2.5. We will make use only of elementary transformations of the type given in Theorem 1.2 combined with Corollary 1.4. We denote by A_j (resp. U_j) the columns of A (resp. U).

1. [Initialize] Set $i \leftarrow n$, $j \leftarrow k$, and let U be the $k \times k$ identity matrix.
2. [Check zero] Set $m \leftarrow j$, and while $m \geq 1$ and $a_{i,m} = 0$, set $m \leftarrow m - 1$. If $m = 0$, the matrix A is not of rank n, so print an error message and terminate the algorithm. Otherwise, if $m < j$, exchange A_m with A_j, $\mathfrak{a}_m$ with $\mathfrak{a}_j$, and U_m with U_j.
3. [Put 1 on the main diagonal] Set $A_j \leftarrow A_j/a_{i,j}$, $U_j \leftarrow U_j/a_{i,j}$, $\mathfrak{a}_j \leftarrow a_{i,j}\mathfrak{a}_j$ and set $m \leftarrow j$. (We now have $a_{i,j} = 1$.)
4. [Loop] If $m = 1$, go to step 6. Otherwise, set $m \leftarrow m - 1$, and if $a_{i,m} = 0$, go to step 4.
5. [Euclidean step] (Here $a_{i,j} = 1$ and $a_{i,m} \neq 0$). Using the algorithm contained in the proof of Theorem 1.2, set $\mathfrak{d} = a_{i,m}\mathfrak{a}_m + \mathfrak{a}_j$ and find $u \in \mathfrak{a}_m\mathfrak{d}^{-1}$ and $v \in \mathfrak{a}_j\mathfrak{d}^{-1}$ such that $a_{i,m}u + v = 1$. Then set $(A_m, A_j) \leftarrow (A_m - a_{i,m}A_j, uA_m + vA_j)$, $(U_m, U_j) \leftarrow (U_m - a_{i,m}U_j, uU_m + vU_j)$ and $(\mathfrak{a}_m, \mathfrak{a}_j) \leftarrow (\mathfrak{a}_m\mathfrak{a}_j\mathfrak{d}^{-1}, \mathfrak{d})$. Finally, go to step 4.
6. [Final Reductions of row i] For $m = j + 1, \dots, m = n$, find $q \in \mathfrak{a}_m\mathfrak{a}_j^{-1}$ such that $a_{i,m} - q$ is small (see below), and set $A_m \leftarrow A_m - qA_j$ and $U_m \leftarrow U_m - qU_j$.
7. [Finished?] If $i = 1$, then output the matrix U, the modified matrix A (i.e., AU in the notation of Theorem 2.5), and the modified ideals $\mathfrak{a}_j$ (i.e., $\mathfrak{b}_j$ in the notation of Theorem 2.5), and terminate the algorithm. Otherwise, set $i \leftarrow i-1$, $j \leftarrow j - 1$ and go to step 2.

Ignoring step 6 for the moment, it is clear that this algorithm, which is essentially identical to the one for $\mathbb{Z}$, terminates with a new matrix A of the form required by Theorem 2.5. Furthermore, the elementary operations that are used are either exchanges of columns (and the corresponding ideals) or transformations allowed by Corollary 1.4, hence the module $\mathfrak{a}_1\omega_1 + \cdots + \mathfrak{a}_k\omega_k$ stays unchanged.

Call $\mathfrak{a}$ the initial ideal product and $\mathfrak{b}$ the current one. All the elementary operations are either of determinant ± 1 (and in that case $\mathfrak{b}$ is unchanged), except in step 3 where the determinant is $1/a_{i,j}$ and $\mathfrak{b}$ is multiplied by $a_{i,j}$, hence the relation $\mathfrak{a} = \det(U)\mathfrak{b}$ is preserved throughout.

Note that upon termination we have a direct sum, and not simply a sum, since the last n columns of A are then linearly independent. This proves Theorem 2.5. We will come back to step 6 of the algorithm after we study uniqueness of HNF pseudobases below. □

Remark. Note that this proof gives an algorithm to find an HNF of a matrix, but does not show that the algorithm is polynomial-time. This is not surprising since the corresponding naïve algorithm for HNF over $\mathbb{Z}$ is not polynomial-time because of coefficient explosion. The existence of a polynomial-time algorithm for HNF reduction (including finding the matrix U) is rather recent (see [5], [7]). Note that in practice, n will be the relative degree of a number field extension, and so in many cases it will be sufficiently small to make the naïve algorithm efficient.

We now consider the problem of uniqueness in Theorem 2.5. We first need a definition.

Definition 2.7. Let (A, I) be a pseudomatrix with $I = (\mathfrak{a}_j)$. If $i_1, \ldots, i_r$ are r distinct rows of A and $j_1, \ldots, j_r$ are r distinct columns, we define the *minor-ideal* corresponding to these indices as follows. Let d be the determinant of the $r \times r$ minor extracted from the given rows and columns of A. Then the minor-ideal is the ideal $d\mathfrak{a}_{j_1} \cdots \mathfrak{a}_{j_r}$.

With this definition we can state:

Theorem 2.8. *With the notation of Theorem* 2.5*, for* $1 \le j \le n$*, set* $\mathfrak{c}_j = \mathfrak{b}_{k-n+j}$*. Then the ideals* $\mathfrak{c}_j$ *are unique. More precisely, if we call* $\mathfrak{g}_j = \mathfrak{g}_j(A)$ *the ideal generated by all the* $(n+1-j) \times (n+1-j)$ *minor-ideals in the last* $n+1-j$ *rows of the matrix* A*, then* $\mathfrak{c}_j = \mathfrak{g}_{n+1-j}\mathfrak{g}_{n-j}^{-1}$*.*

Proof. One easily checks that the ideals $\mathfrak{g}_m(A)$ are invariant under the elementary transformations of the type used in Algorithm 2.6. In particular, $\mathfrak{g}_j(A) = \mathfrak{g}_j(AU)$. But in the last $n+1-j$ rows of AU there is a single nonzero minor whose value is trivially 1, hence we have $\mathfrak{g}_j(A) = \mathfrak{c}_{n+1-j} \cdots \mathfrak{c}_n$, thus proving the theorem. □

Finally, we have the following proposition.

Proposition 2.9. *If* AU *is of the form given by Theorem* 2.5*, a necessary and sufficient condition for* AV *to be of the same form (with the same ideals* $\mathfrak{b}_j$*) is that* $U^{-1}V$ *be a block matrix* $\begin{pmatrix} A & B \\ 0 & D \end{pmatrix}$ *with* D *an* $n \times n$ *upper triangular matrix with* 1 *on the diagonal such that for each* i*,* j *its entry in row* i *and column* j *belongs to* $\mathfrak{c}_i\mathfrak{c}_j^{-1}$*.*

Proof. Trivial and left to the reader. □

Corollary 2.10. *For each i and j with $1 \leq i < j \leq n$, let $S_{i,j}$ be a system of representatives of $K/\mathfrak{c}_i\mathfrak{c}_j^{-1}$. Then in Theorem 2.5 we may assume that for every i and j such that $i < j$, the entry in row i and column j of the matrix AU is in $S_{i,j}$, and in that case the matrix AU is unique.*

Proof. For $i < j$, let $b_{i,j}$ be the entry in row i and column j of the matrix AU. There exists a unique $c_{i,j} \in S_{i,j}$ such that

$$q = c_{i,j} - b_{i,j} \in \mathfrak{c}_i\mathfrak{c}_j^{-1} \ .$$

If the B_j are the columns of AU, then by Proposition 2.9 the replacement of B_j by $B_j - qB_i$ is a legal elementary operation which transforms $b_{i,j}$ into $c_{i,j}$, thus proving the existence. The uniqueness follows also from this, since there was a unique possible q. □

We can now comment on step 6 of Algorithm 2.6. By the above corollary, the reduction done in step 6 is a legal one. Ideally, for each i, j we would like to find a system of representatives of $K/\mathfrak{c}_i\mathfrak{c}_j^{-1}$ as well as an algorithm for finding the representative of a given element of K. There are two different methods of doing this, which both have advantages and disadvantages.

The first method is to compute the (usual) HNF matrix H of $\mathfrak{c}_i\mathfrak{c}_j^{-1}$ on some fixed integral basis of K. If $(d_i)_{1\leq i\leq N}$ are the diagonal elements of H (with $N = [K : \mathbb{Q}]$), then we can take $S = \prod_{1\leq i\leq N} \mathbb{Q}/d_i\mathbb{Z}$ (and as representatives of $\mathbb{Q}/d_i\mathbb{Z}$, for example the interval $[0, d_i)$). If $x \in K$, we express x as a column vector (with rational coefficients) on the integral basis, and then reduce x modulo $\mathfrak{c}_i\mathfrak{c}_j^{-1}$ from bottom to top by subtracting from x suitable multiples of the columns of H so that the coordinates of x fall in the interval $[0, d_i)$ for each i.

We can write this out explicitly as an algorithm.

Algorithm 2.11 (HNF reduction modulo an ideal)**.** Given an ideal $\mathfrak{a}$ by its $N \times N$ upper triangular HNF matrix $H = (h_{i,j})$ in some basis of K, and an element $x \in K$ given by a column vector $X = (x_i)$ in the same basis, this algorithm computes a "canonical" representative of x modulo $\mathfrak{a}$, i.e., an element $y \in K$ such that $x - y \in \mathfrak{a}$ and the coordinates y_i of y in the basis satisfy $0 \leq y_i < h_{i,i}$.

1. [Initialize] Set $i \leftarrow N$, $y \leftarrow x$.
2. [Reduce] Set $q \leftarrow \lfloor y_i/h_{i,i} \rfloor$, $y \leftarrow y - qH_i$ (recall that H_i is the ith column of H).
3. [Finished?] If $i = 1$, output y and terminate the algorithm, otherwise set $i \leftarrow i-1$ and go to step 2.

This method has the advantage of giving a unique and well-defined representative of x modulo $\mathfrak{c}_i\mathfrak{c}_j^{-1}$, as well as an algorithm to find it. However, it will often happen in practice that the first few rows of the HNF matrix H will be very large, and the others much smaller. Hence the resulting "reduced" element will in fact be often quite large.

The second method consists in finding first an LLL-reduced basis L of $\mathfrak{c}_i\mathfrak{c}_j^{-1}$, which will have much smaller coefficients in general than the HNF matrix H. We must then find an element $q \in \mathfrak{c}_i\mathfrak{c}_j^{-1}$ such that $x - q$ is small (we already mentioned the need for this in the remarks following Proposition 1.1). It is well known that this is a difficult problem (probably NP-complete). However, if we write $x = \sum_{1\leq j\leq N} x_j L_j$ with $x_j \in \mathbb{Q}$ (where the L_j are the elements of the basis L) and

choose

$$q = \sum_{1 \le j \le N} \lfloor x_j \rceil L_j$$

(where $\lfloor a \rceil$ denotes one of the nearest integers to a), it is clear that $q \in \mathfrak{c}_i \mathfrak{c}_j^{-1}$ and that $x - q$ is reasonably "small". Note that it is essential that the basis L be LLL-reduced before doing this operation, otherwise $x - q$ would not be small at all in general.

We can write this out explicitly as an algorithm.

Algorithm 2.12 (LLL reduction modulo an ideal)**.** Given an ideal $\mathfrak{a}$ by some $N \times N$ matrix $H = (h_{i,j})$ representing a $\mathbb{Z}$-basis of $\mathfrak{a}$ in some basis of K, and an element $x \in K$ given by a column vector $X = (x_i)$ in the same basis, this algorithm computes a noncanonical but "small" representative of x modulo $\mathfrak{a}$, i.e., an element $y \in K$ such that $x - y \in \mathfrak{a}$ and the coordinates y_i of y in the basis are reasonably small.

1. [LLL-reduce] Using the LLL algorithm or one of its variants, let L be the matrix of an LLL-reduced basis of $\mathfrak{a}$.
2. [Find coefficients] Using Gaussian elimination, find the solution $Z = (z_i)$ to the linear system $LZ = X$ (i.e., $Z = L^{-1}X$)
3. [Reduce] Set $Y \leftarrow X - \sum_{1 \le i \le N} \lfloor z_i \rceil L_i$, output the element y corresponding to Y and terminate the algorithm.

The main advantage of this method is that the reduced vector will have much smaller entries. However, the reduction is not unique, and takes more time since LLL is usually slower than HNF. Only practice can tell which method is to be preferred. In the modular HNF method explained below, however, it is essential to use this method to avoid coefficient explosion.

The above algorithm can be considerably improved by using an idea explained to me by Peter Montgomery. Instead of doing an LLL reduction of the ideal, which is an expensive operation, we can perform a fast *partial* reduction of the matrix (a matrix A with columns A_j will be said to be partially reduced if for any distinct columns we have $\|A_i \pm A_j\| \ge \|A_j\|$.)

The resulting basis will not be LLL-reduced in general, but its entries will be of comparable size to that of the LLL-reduced one. Furthermore, it is particularly well suited to matrices which have a few rows much larger than the others, such as typical HNF matrices for ideals. I refer to [8] for details.

It is necessary to make a number of remarks concerning the implementation of the HNF algorithm in Dedekind domains (Algorithm 2.6).

Usually a torsion-free R-module M will be given by a generating set expressed in a fixed basis $\mathcal{B}$ of KM. Using Algorithm 2.6, we can find a pseudobasis $(\omega_j, \mathfrak{a}_j)_{1 \le j \le n}$ which will have the very special property of being upper triangular with ones on the diagonal when expressed in $\mathcal{B}$.

We can now start modifying this basis. First we can choose to have only integral (and even primitive) ideals $\mathfrak{a}_j$ by dividing them by suitable elements of $\mathbb{Q}^*$, and multiplying the corresponding ω_j by the same. Alternatively, we can ask for integral coefficients for the ω_j, and this is done in a similar manner.

Then we can ask for a pseudobasis such that all the ideals are equal to R except perhaps the last, whose ideal class will then be the Steinitz class of M. That this

is possible follows from Proposition 1.11 together with Corollary 1.5. By induction, we can replace ideal pairs $(\mathfrak{a}_j, \mathfrak{a}_{j+1})$ by $(R, \mathfrak{a}_j\mathfrak{a}_{j+1})$ by using legal elementary transformations on the matrix A, and hence at the end of the process all ideals except perhaps the last one will be equal to R, as desired. Note however that to apply Proposition 1.11 in an algorithmic manner, it is necessary to know the prime decompositions of the norms of the $\mathfrak{a}_j$. In practice, this is always the case, but of course in general this is perhaps not a polynomial-time operation. Furthermore, note that ultimately Proposition 1.11 relies on being able to find an integral ideal in a given ideal class coprime to some other ideal, which can be done deterministically only with the approximation theorem, hence can be rather slow.

Finally, note that if we perform the above transformations on the matrix and the ideals, the resulting pseudobasis will no longer be represented by a triangular matrix. If we are still not content with this, we could, if desired, obtain an $(n+1)$-element generating set of our module by replacing $\omega_n\mathfrak{a}_n$ with $a\omega_n + b\omega_n$, where $\mathfrak{a}_n = aR + bR$ is found using Proposition 1.9. This will of course not be a direct sum. Note that the search for a and b can be done in polynomial time if the norm of $\mathfrak{a}_n$ is completely factored, since a can be taken equal to the norm of $\mathfrak{a}_n$. We may also like to know if our module M is free and find a basis. Using the techniques developed in [2, Chapter 6], once a relation matrix is found which is sufficient to compute the class group and regulator of R, it is quite easy to determine whether an ideal is principal or not, and if it is, to find a generator. Note that [2] assumes the GRH, but evidently the same technique applies as long as we have obtained a relation matrix.

So we test if $\mathfrak{a}_n$ is a principal ideal. If it is not, then nothing more can be done: according to Theorem 2.1, M is not free, so either use the pseudobasis (probably the best), or the $(n+1)$-element generating set. If $\mathfrak{a}_n = aR$, then after replacing ω_n by $a\omega_n$, $(\omega_j)_{1\le j\le n}$ is an R-basis of M.

If we only want to know whether M is free or not, without finding explicitly a basis, then it is not necessary to use Proposition 1.11 inductively: we use the initial HNF pseudobasis and test whether $\mathfrak{a}_1 \dots \mathfrak{a}_n$ is a principal ideal or not.

3. The modular Hermite Normal Form algorithm over Dedekind domains

It is well known that the usual HNF over $\mathbb{Z}$ suffers from coefficient explosion, which often makes the algorithm quite impractical, even for matrices of reasonable size. Since our algorithm is a direct generalization of the naïve HNF algorithm, the same phenomenon occurs. Hence, it is necessary to improve the basic algorithm.

In the case of the ordinary HNF, there are essentially two ways of doing this, depending on what one wants.

The first method is the "modular" method (see [5], [7]). If we can compute the determinant of the lattice generated by the columns of our matrix, all computations can then be done modulo this determinant, and the final HNF matrix can be recovered by a simple GCD procedure (see [2, Algorithm 2.4.6]). This method can be proved to be polynomial-time, but has the disadvantage of not computing the (unimodular) transformation matrix U. In most cases, this is not needed anyway, but in other cases it is essential (see for example the proof of Proposition 1.1). If we really want the matrix U, it can be recovered from the modular method, but its

coefficients will in general be huge and the method would not be polynomial-time. However, see [5] for a way of obtaining U in polynomial time.

The other methods, due essentially to Havas (see [6] and the references therein), are more heuristic in nature (they are not provably polynomial-time), but have the great advantage of giving very small transformation matrices U. Since in our application to relative extensions of number fields we will usually not need the matrix U, we will not consider here the generalization of Havas's algorithms to the Dedekind case, although there is no doubt that it can be done. Hence, the purpose of this section is to explain how the usual modular HNF algorithm can be modified to work over Dedekind domains. Although quite simple, this generalization is not absolutely straightforward, so we give some details, following closely the exposition of [5] and [2].

We have defined above the notion of minor-ideal of a pseudomatrix (A, I), and in particular $\mathfrak{g}_1(M)$ is the ideal of R generated by all $n \times n$ minor-ideals of the pseudomatrix (A, I). We will say that $\mathfrak{g}_1(M)$ is the *determinantal ideal* of the module M. It is clearly a generalization of the notion of determinant of a lattice.

Since there are $\binom{k}{n}$ minors of order n, it could be a lengthy task to compute $\mathfrak{g}_1(M)$ explicitly, except of course when $k = n$ or even $k = n + 1$ (note that the computation of each minor is an ordinary determinant computation which can be done with the usual Gauss-Bareiss pivoting strategy, which only involves exact divisions).

However, we do not really need the determinantal ideal itself but only an integral multiple of it. Furthermore, if we choose $n - 1$ fixed independent columns, and consider the $k - n + 1$ order-n minors obtained by choosing successively each of the remaining columns, we have a much more reasonable number of minor-ideals to compute, their computation is very fast (since $n - 1$ of the pivoting steps are done once and for all), and the ideal sum of all these minor-ideals will give a reasonably sized multiple of the determinantal ideal $\mathfrak{g}_1(M)$. Hence, we may assume that we have computed an ideal $\mathfrak{D}$ which is an integral multiple (i.e., is a subset) of the determinantal ideal $\mathfrak{g}_1(M)$ of M. We now describe what modifications must be made to Algorithm 2.6. We will make the computations in this algorithm modulo $\mathfrak{D}$, and then we will have to recover the correct HNF pseudomatrix by suitable ideal operations.

First, we must compute modulo $\mathfrak{D}$. Recall that the individual columns A_j or ideals $\mathfrak{a}_j$ are quite arbitrary, and that only the rank-1 submodule $\mathfrak{a}_j A_j$ of M is a reasonable object to consider. Hence, we must reduce modulo $\mathfrak{D}$ not the column A_j itself, but the module $\mathfrak{a}_j A_j$. In other words, we must reduce the column A_j modulo the ideal $\mathfrak{D}\mathfrak{a}_j^{-1}$.

Hence, we will modify step 5 of Algorithm 2.6 as follows. Before returning to step 4, we will set $A_m \leftarrow A_m \pmod{\mathfrak{D}\mathfrak{a}_m^{-1}}$ and $A_j \leftarrow A_j \pmod{\mathfrak{D}\mathfrak{a}_j^{-1}}$. Here, the reduction modulo an ideal is understood in the sense of the LLL-reduction Algorithm 2.12.

Since in the inner loop of Algorithm 2.6 the column index j is fixed and only m varies, it can also be argued that we should only perform the reduction of the column A_m, and perform the reduction of A_j only when the m-loop is finished. Although this avoids almost half of the (expensive) reductions, it may lead to much larger intermediate coefficients, so it is not clear if this method is preferable. Once this modified algorithm is finished, we must execute the following supplementary algorithm to recover the true HNF pseudobasis of M (see [2], [5]).

Algorithm 3.1 (Modular HNF algorithm in Dedekind domains)**.** Given an $n \times k$ matrix $A = (a_{i,j})$ of rank n, and k (fractional) ideals $\mathfrak{a}_j$ in a number field K, this algorithm computes an HNF pseudobasis (W, I) of the module $M = \sum_j \mathfrak{a}_j A_j$, where W is an $n \times n$ upper triangular matrix with ones on the diagonal, and $I = (\mathfrak{b}_1, \ldots, \mathfrak{b}_n)$ is a list of n ideals. We assume that we have computed a multiple $\mathfrak{D}$ of the determinantal ideal of M.

1. [Compute HNF modulo $\mathfrak{D}$] Using Algorithm 2.6 above, together with the modifications that we have just described for working modulo $\mathfrak{D}$, let $B = (b_{i,j})$ be the $n \times n$ HNF matrix obtained by discarding the first $k - n$ zero-columns from the resulting matrix AU, and let $\mathfrak{b}_j$ be the corresponding ideals (we discard in Algorithm 2.6 all the statements concerning the matrix U). Then set $\mathfrak{B} \leftarrow \mathfrak{D}$, $i \leftarrow n$.
2. [Euclidean step] Set $\mathfrak{d} = b_{i,i}\mathfrak{b}_i + \mathfrak{B}$, and using Theorem 1.2, find $u \in \mathfrak{b}_i\mathfrak{d}^{-1}$ and $v \in \mathfrak{B}\mathfrak{d}^{-1}$ such that $b_{i,i}u + v = 1$. Then set $W_i \leftarrow uB_i \pmod{\mathfrak{B}\mathfrak{d}^{-1}}$ and $\mathfrak{b}_i \leftarrow \mathfrak{d}$ (here again reduction is done using Algorithm 2.12). Set $w_{i,i} \leftarrow 1$. (Note that $ub_{i,i} \equiv 1 \pmod{\mathfrak{B}\mathfrak{d}^{-1}}$ but the reduction modulo $\mathfrak{B}\mathfrak{d}^{-1}$ may not reduce it to 1.)
3. [Finished?] If $i > 1$, set $\mathfrak{B} \leftarrow \mathfrak{B}\mathfrak{d}^{-1}$ and go to step 2. Otherwise, for $i = n - 1$, $n - 2$, ... , 1, and for $j = i + 1$, ... , n, using Algorithm 2.12, find $q \in \mathfrak{b}_i\mathfrak{b}_j^{-1}$ such that $w_{i,j} - q$ is small, and set $W_j \leftarrow W_j - qW_i$. Output the matrix W, the ideal list $I = (\mathfrak{b}_1, \ldots, \mathfrak{b}_n)$ and terminate the algorithm.

Proof. The proof of the validity of this algorithm is essentially the same as in the classical case (see [2] and [5]), and for brevity's sake we do not repeat it here. The $\mathfrak{g}_i(A)$ which is defined in the classical case as the GCD of all $i \times i$ minors extracted from the last i rows of A is replaced in our situation by the minor-ideal $\mathfrak{g}_i(M)$ which plays exactly the same role (and reduces to the classical definition in the case where $\mathbb{Z}_K = \mathbb{Z}$). Note that, according, for example, to Proposition 1.3 (see also the remark after Corollary 1.5), the elementary column transformations made in step 3 are legal. □

We finish this section by noting that it is more efficient in practice to interleave Algorithm 2.6 and Algorithm 3.1 into a single algorithm, analogous to [2, Algorithm 2.4.8] (see also [4]). The proof of the validity of this algorithm follows from the proofs given above.

Algorithm 3.2 (Modular HNF Algorithm in Dedekind domains)**.** Given an $n \times k$ matrix $A = (a_{i,j})$ of rank n, and k (fractional) ideals $\mathfrak{a}_j$ in a number field K, this algorithm computes an HNF pseudobasis (W, I) of the module $M = \sum_j \mathfrak{a}_j A_j$, where W is an $n \times n$ upper triangular matrix with 1 on the diagonal, and $I = (\mathfrak{b}_1, \ldots, \mathfrak{b}_n)$ is a list of n ideals. We assume that we have computed a multiple $\mathfrak{D}$ of the determinantal ideal of M.

1. [Initialize] Set $i \leftarrow n$, $j \leftarrow k$ and $\mathfrak{B} \leftarrow \mathfrak{D}$.
2. [Check zero] Set $m \leftarrow j$, and while $m \geq 1$ and $a_{i,m} = 0$, set $m \leftarrow m - 1$. (Note that since we know that $\mathfrak{D}$ is a nonzero ideal, it is not necessary to check that the matrix A is of maximal rank.) If $m < j$, exchange A_m with A_j and $\mathfrak{a}_m$ with $\mathfrak{a}_j$.
3. [Put 1 on the main diagonal] Set $A_j \leftarrow A_j/a_{i,j}$, $\mathfrak{a}_j \leftarrow a_{i,j}\mathfrak{a}_j$ and set $m \leftarrow j$. (We now have $a_{i,j} = 1$.)
4. [Loop] If $m = 1$, go to step 6. Otherwise, set $m \leftarrow m - 1$, and if $a_{i,m} = 0$, go to step 4.

5. [Euclidean step] (Here $a_{i,j} = 1$ and $a_{i,m} \neq 0$). Using the algorithm contained in the proof of Theorem 1.2, set $\mathfrak{d} = a_{i,m}\mathfrak{a}_m + \mathfrak{a}_j$ and find $u \in \mathfrak{a}_m\mathfrak{d}^{-1}$ and $v \in \mathfrak{a}_j\mathfrak{d}^{-1}$ such that $a_{i,m}u + v = 1$. Then set in this order $(A_m, A_j) \leftarrow (A_m - a_{i,m}A_j, uA_m + vA_j)$, $(\mathfrak{a}_m, \mathfrak{a}_j) \leftarrow (\mathfrak{a}_m\mathfrak{a}_j\mathfrak{d}^{-1}, \mathfrak{d})$, $A_m \leftarrow A_m \pmod{\mathfrak{B}\mathfrak{a}_m^{-1}}$ and $A_j \leftarrow A_j \pmod{\mathfrak{B}\mathfrak{a}_j^{-1}}$, where the reduction is done using Algorithm 2.12. Finally, go to step 4.
6. [Next row] Set $\mathfrak{d} \leftarrow a_{i,j}\mathfrak{a}_j + \mathfrak{B}$ and using Theorem 1.2 once again compute $u \in \mathfrak{a}_j\mathfrak{d}^{-1}$ and $v \in \mathfrak{B}\mathfrak{d}^{-1}$ such that $ua_{i,j} + v = 1$. Set $W_i \leftarrow uA_j \pmod{\mathfrak{B}\mathfrak{d}^{-1}}$ (where the reduction is again done using Algorithm 2.12), $\mathfrak{a}_i \leftarrow \mathfrak{d}$ and set $w_{i,i} \leftarrow 1$. For $m = j+1, \ldots, m = n$, using Algorithm 2.12 once more, find $q \in \mathfrak{a}_m\mathfrak{a}_j^{-1}$ such that $a_{i,m} - q$ is small, and set $A_m \leftarrow A_m - qA_j$.
7. [Finished?] If $i = 1$, then output the matrix W, the modified ideals $\mathfrak{a}_j$ and terminate the algorithm. Otherwise, set $\mathfrak{B} \leftarrow \mathfrak{B}\mathfrak{d}^{-1}$, $i \leftarrow i-1$, $j \leftarrow j-1$ and go to step 2.

Remark. The above modular version performs well in practice. It seems quite plausible that, as in the case of $R = \mathbb{Z}$, this algorithm is in fact polynomial-time, but I have not tried to prove this, although it may be easy.

4. The Smith Normal Form algorithm over Dedekind domains

Recall the elementary divisor theorem for torsion-free modules.

Theorem 4.1. *Let P and N be two torsion-free modules of rank p and n, respectively, such that $N \subset P$ (so $n \leq p$). There exist fractional ideals $\mathfrak{b}_1, \ldots, \mathfrak{b}_p$ of R, a basis $(\omega_1, \ldots, \omega_p)$ of $V = PK$ and integral ideals $\mathfrak{d}_1, \ldots, \mathfrak{d}_n$ such that*

$$P = \mathfrak{b}_1\omega_1 \oplus \cdots \oplus \mathfrak{b}_p\omega_p \quad \text{and} \quad N = \mathfrak{d}_1\mathfrak{b}_1\omega_1 \oplus \cdots \oplus \mathfrak{d}_n\mathfrak{b}_n\omega_n$$

and such that $\mathfrak{d}_{i-1} \subset \mathfrak{d}_i$ for $2 \leq i \leq n$.

The ideals $\mathfrak{d}_i$ (for $1 \leq i \leq n$) and the ideal classes *of the ideal products $\mathfrak{b}_1 \cdots \mathfrak{b}_n$ and $\mathfrak{b}_{n+1} \cdots \mathfrak{b}_p$ depend only on P and N.*

In other words, this theorem says that we can find pseudobases of P and N which differ only in their ideals, in a specific way. Our main goal will be to give an algorithm to find these pseudobases. This will be the Smith Normal Form algorithm.

Before doing this, we must generalize the notion of pseudomatrix. If (A, I) is a pseudomatrix with $A = (a_{i,j})$ an $n \times k$ matrix with coefficients in K and $I = (\mathfrak{a}_i)$ a vector of k ideals, it is natural to consider the linear map f from $\mathfrak{a}_1 \times \cdots \times \mathfrak{a}_k$ to K^n associated with this pseudomatrix, defined by

$$f(a_1, \ldots, a_k) = \sum_{1 \leq j \leq k} a_j A_j \ ,$$

where as usual A_j denotes the jth column of A, considered as an element of K^n. The image of this map f is exactly the module $M = \sum_j \mathfrak{a}_j A_j$ with which we have worked.

We must now consider the more general situation where the map f is a linear map from $N = \mathfrak{a}_1 \times \cdots \times \mathfrak{a}_n$ to $P = \mathfrak{b}_1 \times \cdots \times \mathfrak{b}_p$ for some other ideals $\mathfrak{b}_i$. If we call i_j the jth canonical injection from $\mathfrak{a}_j$ to N (defined by $i_j(a) = (0, \ldots, 0, a, 0, \ldots, 0)$ where a is at the jth component) and p_i the ith canonical projection of P to $\mathfrak{b}_i$ (defined by $p_i(b_1, \ldots, b_n) = b_i$), we will set

$$f_{i,j} = p_i \circ f \circ i_j \ .$$

This is a linear map from $\mathfrak{a}_j$ to $\mathfrak{b}_i$. Conversely, given any linear maps $g_{i,j}$ from $\mathfrak{a}_j$ to $\mathfrak{b}_i$, we can define in a unique manner a linear map f from N to P such that $f_{i,j} = g_{i,j}$.

Now let $\mathfrak{a}$ and $\mathfrak{b}$ be two ideals and g a linear map from $\mathfrak{a}$ to $\mathfrak{b}$. By tensoring with the field K we can extend this to a map (which we denote again by g) from K to K (since $\mathfrak{a}$ and $\mathfrak{b}$ are nonzero fractional ideals), and such a map is of the form $g(x) = \lambda x$ for some $\lambda \in K$. Conversely, such a λ gives a map from $\mathfrak{a}$ to $\mathfrak{b}$ if and only if $\lambda\mathfrak{a} \subset \mathfrak{b}$, i.e., $\lambda \in \mathfrak{b}\mathfrak{a}^{-1}$. This leads us to the following definition.

Definition and Proposition 4.2. *Let $N = \mathfrak{a}_1\omega_1 \oplus \cdots \oplus \mathfrak{a}_n\omega_n$ and $P = \mathfrak{b}_1\eta_1 \oplus \cdots \oplus \mathfrak{b}_p\eta_p$ be two torsion-free R-modules given by pseudobases, and let $A = (a_{i,j})$ be a $p \times n$ matrix. Let $I = (\mathfrak{b}_1, \ldots, \mathfrak{b}_p)$ and $J = (\mathfrak{a}_1, \ldots, \mathfrak{a}_n)$.*

(1) *We will say that (A, I, J) is an* integral *pseudomatrix if for each i and j we have $a_{i,j} \in \mathfrak{b}_i\mathfrak{a}_j^{-1}$.*

(2) *Given such a pseudomatrix (A, I, J), the map f from N to P associated with it is the map defined by setting*

$$f(\sum_j a_j\omega_j) = \sum_j a_j f(\omega_j) = \sum_j a_j \sum_i a_{i,j}\eta_i = \sum_i \eta_i(\sum_j a_{i,j}a_j) \ ,$$

which makes sense since $a_{i,j}a_j \in \mathfrak{b}_i$.

(3) *The module M associated with (A, I, J) is the quotient module*

$$P/f(N) = (\mathfrak{b}_1\eta_1 \oplus \cdots \oplus \mathfrak{b}_p\eta_p)/f(\mathfrak{a}_1\omega_1 \oplus \cdots \oplus \mathfrak{a}_n\omega_n) \ .$$

Note that the module M associated with a pseudomatrix (A, I, J) is a torsion module if and only if $p = n$, i.e., if A is a square matrix (of nonzero determinant).

We can now state the main theorem of this section. For simplicity we state it for square matrices, but it is easily extended to the general case.

Theorem 4.3 (Smith Normal Form in Dedekind domains)**.** *Let (A, I, J) be a pseudomatrix as above, with $A = (a_{i,j})$ an $n \times n$ matrix and $I = (\mathfrak{b}_1, \ldots, \mathfrak{b}_n)$, and $J = (\mathfrak{a}_1, \ldots, \mathfrak{a}_n)$ two vectors of n ideals such that $a_{i,j} \in \mathfrak{b}_i\mathfrak{a}_j^{-1}$.*

Then there exist vectors of ideals $(\mathfrak{b}'_1, \ldots, \mathfrak{b}'_n)$ and $(\mathfrak{a}'_1, \ldots, \mathfrak{a}'_n)$, and two $n \times n$ matrices $U = (u_{i,j})$ and $V = (v_{i,j})$ satisfying the following conditions. For all i, set $\mathfrak{d}_i = \mathfrak{a}'_i{\mathfrak{b}'_i}^{-1}$, $\mathfrak{a} = \mathfrak{a}_1 \cdots \mathfrak{a}_n$, $\mathfrak{b} = \mathfrak{b}_1 \cdots \mathfrak{b}_n$, $\mathfrak{a}' = \mathfrak{a}'_1 \cdots \mathfrak{a}'_n$ and $\mathfrak{b}' = \mathfrak{b}'_1 \cdots \mathfrak{b}'_n$. Then

(1) $\mathfrak{a} = \det(U)\mathfrak{a}'$ *and* $\mathfrak{b}' = \det(V)\mathfrak{b}$ (*note the reversal*) .

(2) *The matrix VAU is the $n \times n$ identity matrix.*

(3) *The $\mathfrak{d}_i$ are integral ideals and for $2 \le i \le n$ we have $\mathfrak{d}_{i-1} \subset \mathfrak{d}_i$.*

(4) *For all i, j we have $u_{i,j} \in \mathfrak{a}_i{\mathfrak{a}'_j}^{-1}$ and $v_{i,j} \in \mathfrak{b}'_i\mathfrak{b}_j^{-1}$.*

Proof. Again we prove this theorem by giving an explicit algorithm for contructing the Smith normal form. We follow closely [2, Algorithm 2.4.14], except that we do not work modulo the determinant (although of course such a modular version of the Smith Normal Form algorithm is easily written).

Algorithm 4.4 (SNF algorithm in Dedekind domains)**.** Given an invertible $n \times n$ matrix $A = (a_{i,j})$, and two lists of n (fractional) ideals $I = (\mathfrak{b}_i)$ and $J = (\mathfrak{a}_j)$ in a number field K, this algorithm computes two other lists of n ideals $\mathfrak{b}'_i$ and $\mathfrak{a}'_j$ and two $n \times n$ matrices U and V such that this data satisfies the conditions of

Theorem 4.3. We will make use only of elementary transformations of the type given in Theorem 1.2 combined with Corollary 1.4. We denote by A_j (resp. U_j) the columns of A (resp. U), and by A'_j (resp. V'_j) the rows of A (resp. V).

1. [Initialize i] Set $i \leftarrow n$, and let U and V be the $n \times n$ identity matrix. If $n = 1$, output $\mathfrak{b}_1$, $\mathfrak{a}_1$, U, V and terminate the algorithm.
2. [Initialize j for row reduction] Set $j \leftarrow i$ and $c \leftarrow 0$.
3. [Check zero] If $j = 1$, go to step 5. Otherwise, set $j \leftarrow j - 1$. If $a_{i,j} = 0$ go to step 3.
4. [Euclidean step] Using the algorithm of Theorem 1.2, set $\mathfrak{d} \leftarrow a_{i,i}\mathfrak{a}_i + a_{i,j}\mathfrak{a}_j$ and find $u \in \mathfrak{a}_i\mathfrak{d}^{-1}$ and $v \in \mathfrak{a}_j\mathfrak{d}^{-1}$ such that $a_{i,i}u + a_{i,j}v = 1$. Then set $(A_j, A_i) \leftarrow (a_{i,j}A_j - a_{i,i}A_i, uA_i + vA_j)$, $(U_j, U_i) \leftarrow (a_{i,j}U_j - a_{i,i}U_i, uU_i + vU_j)$, $(\mathfrak{a}_j, \mathfrak{a}_i) \leftarrow (\mathfrak{a}_i\mathfrak{a}_j\mathfrak{d}^{-1}, \mathfrak{d})$. Finally, go to step 3.
5. [Initialize j for column reduction] Set $j \leftarrow i$, and if $a_{i,i} \neq 1$, set $U_i \leftarrow U_i/a_{i,i}$, $\mathfrak{a}_i \leftarrow a_{i,i}\mathfrak{a}_i$, $a_{i,i} \leftarrow 1$.
6. [Check zero] If $j = 1$, go to step 8. Otherwise, set $j \leftarrow j - 1$. If $a_{j,i} = 0$ go to step 6.
7. [Euclidean step] Using the algorithm of Theorem 1.2, set $\mathfrak{d} \leftarrow \mathfrak{b}_i^{-1} + a_{j,i}\mathfrak{b}_j^{-1}$ and find $u \in \mathfrak{b}_i^{-1}\mathfrak{d}^{-1}$ and $v \in \mathfrak{b}_j^{-1}\mathfrak{d}^{-1}$ such that $u + a_{j,i}v = 1$. Then set $(A'_j, A'_i) \leftarrow (a_{j,i}A'_j - A'_i, uA'_i + vA'_j)$, $(V'_j, V'_i) \leftarrow (a_{j,i}V'_j - V'_i, uV'_i + vV'_j)$, $(\mathfrak{b}_j, \mathfrak{b}_i) \leftarrow (\mathfrak{b}_i\mathfrak{b}_j\mathfrak{d}, \mathfrak{d}^{-1})$. Finally set $c \leftarrow c + 1$ and go to step 6.
8. [Repeat stage i?] If $c > 0$, go to step 2.
9. [Check the rest of the matrix] Set $\mathfrak{b} \leftarrow \mathfrak{a}_i\mathfrak{b}_i^{-1}$. For $1 \leq k, l < i$ check whether $a_{k,l}\mathfrak{a}_l\mathfrak{b}_k^{-1} \subset \mathfrak{b}$. As soon as this is not the case, let $\mathfrak{d} \leftarrow \mathfrak{b}_i\mathfrak{b}_k^{-1}$. Let d be one of the generators of $\mathfrak{d}$ such that $a_{k,l}d \notin \mathfrak{a}_i\mathfrak{a}_l^{-1}$ (such a generator must exist and is easy to find, for example by looking at the $\mathbb{Z}$-basis of $\mathfrak{d}$ given by the ordinary HNF). Set $A'_i \leftarrow A'_i + dA'_k$, $V'_i \leftarrow V'_i + dV'_k$ and go to step 2.
10. [Next stage] (Here $a_{k,l}\mathfrak{a}_l\mathfrak{b}_k^{-1} \subset \mathfrak{b}$ for all $k, l < i$). If $i \geq 3$, set $i \leftarrow i - 1$ and go to step 2. Otherwise, set $U_1 \leftarrow U_1/a_{1,1}$, $\mathfrak{a}_1 \leftarrow a_{1,1}\mathfrak{a}_1$ and $a_{1,1} \leftarrow 1$, output the matrices U and V, the two ideal lists $(\mathfrak{b}_i)$ and $(\mathfrak{a}_j)$ and terminate the algorithm.

Contrary to the HNF algorithm which was immediate, there are several things to be checked. First we must check that this algorithm is valid. It is easily verified that all the elementary operations that are used are legal ones and that the identities $\mathfrak{a} = \det(U)\mathfrak{a}'$ and $\mathfrak{b}' = \det(V)\mathfrak{b}$ are preserved throughout. Furthermore, upon termination the matrix A will be the identity matrix and we will have $a_{j,j}{\mathfrak{b}'_j}^{-1}\mathfrak{a}'_j \subset a_{i,i}{\mathfrak{b}'_i}^{-1}\mathfrak{a}'_i$ for all $j < i$, hence since $a_{i,i} = a_{j,j} = 1$, we obtain from the definition of the $\mathfrak{d}_i$ that $\mathfrak{d}_j \subset \mathfrak{d}_i$ for all $j < i$. In addition, it is easily checked that the ideal $\mathfrak{c} = \sum_{i,j} a_{i,j}\mathfrak{a}_j\mathfrak{b}_i^{-1}$ is preserved by all the elementary transformations of rows and columns that we perform. Since we have assumed that $a_{i,j} \in \mathfrak{b}_i\mathfrak{a}_j^{-1}$ it follows that $\mathfrak{c}$ is an integral ideal. But on the final pseudomatrix we have $\mathfrak{c} = \sum_i \mathfrak{a}'_i\mathfrak{b}'_i = \sum_i \mathfrak{d}_i = \mathfrak{d}_n$ since $\mathfrak{d}_j \subset \mathfrak{d}_i$ for $j < i$. It follows that $\mathfrak{d}_n$ is an integral ideal, hence all the $\mathfrak{d}_i$ are integral ideals.

Note that we could interpret all the $\mathfrak{d}_i$ in the same way by taking the sum of *all* $(n-i) \times (n-i)$ minor-ideals of the pseudomatrix.

We must now show that the algorithm terminates. First note that the effect of steps 2 to 8 on the triplet $(a_{i,i}, \mathfrak{a}_i, \mathfrak{b}_i^{-1})$ is to transform it into

$$(1, \sum_{j \leq i} a_{i,j}\mathfrak{a}_j, \mathfrak{b}_i^{-1} + \sum_{j<i} a'_{j,i}\mathfrak{b}_j^{-1}) \ ,$$

where the $a'_{j,i}$ are the coefficients of the matrix after step 4. Hence, the product $a_{i,i}\mathfrak{a}_i\mathfrak{b}_i^{-1}$, which is an integral ideal throughout the algorithm (since it is included in the ideal $\mathfrak{c} = \mathfrak{d}_n$) can only get larger. Now since all the ideals are nonzero, steps 2 to 8 can leave this product unchanged only if $a_{i,j} = 0$ and $a'_{j,i} = 0$ for all $j < i$, and this implies that $c = 0$, which is the termination condition of the loop from steps 2 to 8. Thus, we have a strictly increasing sequence of integral ideals, which is thus finite. So we get to step 9 after a finite number of steps.

Now one loop from step 9 back to step 5 again transforms the triplet $(1, \mathfrak{a}_i, \mathfrak{b}_i^{-1})$ into

$$(1, \mathfrak{a}_i + \sum_{j<i} da_{k,j}\mathfrak{a}_j, \mathfrak{b}_i^{-1})$$

and hence, since $da_{k,l} \notin \mathfrak{a}_i\mathfrak{a}_l^{-1}$, it follows that the new ideal $\mathfrak{a}_i$ is strictly larger, and hence the new $a_{i,i}\mathfrak{a}_i\mathfrak{b}_i^{-1}$ also. We again have a strictly increasing sequence of integral ideals, which is thus finite, hence we pass only a finite number of times through step 9, so the algorithm terminates. □

Remarks.

(1) Considering step 7 of the algorithm, in practice it will probably be better to keep the ideals $\mathfrak{b}_i^{-1}$ and not the ideals $\mathfrak{b}_i$ themselves, so as to diminish the number of ideal inversions.

(2) As mentioned earlier, it is very easy to introduce a modular version of the SNF algorithm, as in Algorithm 2.4.14 of [2]. Such a variant is necessary in many cases to avoid coefficient explosion. In addition, the algorithm is easily modified to deal with singular or nonsquare matrices.

(3) Note that the module M associated with the pseudomatrix (A, I, J) will be isomorphic to

$$R/\mathfrak{d}_1 \oplus \cdots \oplus R/\mathfrak{d}_n \ ,$$

and thus this gives the complete structure of M as an R-module.

References

1. W. Bosma and M. Pohst, *Computations with finitely generated modules over Dedekind rings*, Proceedings ISSAC'91 (1991), 151–156.
2. H. Cohen, *A course in computational algebraic number theory*, Graduate Texts in Math., vol. 138, Springer-Verlag, 1993. MR **94i:**11105
3. H. Cohen, F. Diaz y Diaz and M. Olivier, *Algorithmic computations in relative extensions of number fields*, in preparation.
4. P. D. Domich, R. Kannan and L. E. Trotter, Jr., *Hermite normal form computation using modulo determinant arithmetic*, Math. Oper. Research **12** (1987), 50–59. MR **88e:**65047
5. J. Hafner and K. McCurley, *Asymptotically fast triangularization of matrices over rings*, SIAM J. Comput. **20** (1991), 1068–1083. MR **93d:**15021
6. G. Havas and B. Majewski, *Hermite normal form computation for integer matrices*, Congr. Numer. **105** (1994), 184–193. CMP 96:10
7. R. Kannan and A. Bachem, *Polynomial algorithms for computing the Smith and Hermite normal form of an integer matrix*, SIAM J. Comput. **8** (1979), 499-507. MR **81k:**15002
8. P. Montgomery, *in preparation.*

Laboratoire A2X, UMR 9936 du C.N.R.S., Université Bordeaux I, 351 Cours de la Libération, 33405 Talence Cedex, France

E-mail address: cohen@math.u-bordeaux.fr