研究レポート
二次体における効率的な素イデアル分解のアルゴリズム

横浜市立南高等学校 2 年　伊藤充洋

研究期間：2021 年 6 月～2021 年 9 月

1　研究目的

整数における素因数分解の一意性は，非常に重要な概念である．例えば，有理数の 2 乗でない有理数の平方根は無理数である，といった事実は素因数分解の一意性から簡単に導ける．一方，整数において成り立つ素因数分解の一意性は，整数より広い世界では必ずしも成り立つとは限らない．例えば，有名な例として，整数 a, b により $a + b\sqrt{-5}$ とかける数全体の集合，すなわち $\mathbb{Q}(\sqrt{-5})$ の整数環では「素因数分解の一意性」が成り立たない．実際，$6 = 2 \cdot 3 = (1 + \sqrt{-5})(1 - \sqrt{-5})$ と，6 は二通りに「素因数分解」される．そこで，整数より広い世界でも素因数分解をするために，整数論では「イデアル」や「素イデアル」という概念を考える．これらはそれぞれ，整数における「整数の倍数全体の集合」と「素数の倍数全体の集合」の一般化になっている．上記の $\mathbb{Q}(\sqrt{-5})$ のような代数体の整数環においては素因数分解に対応する「素イデアル分解」の一意性が成り立ち，この性質は整数論の様々な問題を解く上で有用な道具として知られている．

例えば $p = \pm(x^2 - dy^2)$ ($d \equiv 2, 3 \pmod 4$ は平方因子を持たず，p は素数）という形の不定方程式に対しては，$\mathbb{Q}(\sqrt{d})$ の整数環，つまり $\mathbb{Z}[\sqrt{d}]$ における考察が非常に役に立つ．右辺は $\pm(x - y\sqrt{d})(x + y\sqrt{d})$ となり，例えば p が $\mathbb{Z}[\sqrt{d}]$ において生成するイデアル $\langle p \rangle$ の素イデアル分解の議論に持ち込むことができる．

また，1 の原始 p 乗根 ζ_p を $\mathbb{Q}$ に添加した $\mathbb{Q}(\zeta_p)$ の整数環における議論により，正則素数と呼ばれる特殊な素数 p に対してフェルマーの最終定理の $n = p$ の場合を示すことができる．これは，$z^p = (x + y)(x + \zeta_p y) \dots (x + \zeta_p^{p-1} y)$ と「因数分解」し，素イデアル分解の一意性を用いて右辺の各因子の生成するイデアルが互いに素なイデアルの p 乗になる場合に帰着し ... というように議論が展開される．

このように，代数体の整数環におけるイデアルなどの計算は，整数論の問題を解く上で極めて重要なものである．しかし，イデアルの計算を行うには膨大な量の計算を行わなければならず，手計算でこれを実行するのは現実的に難しいことが多い．そこで本論文では，特に整数環が簡単に決定できる二次体を対象に，イデアルの素イデアル分解をはじめとした様々な計算アルゴリズムについて研究することにした．具体的な素イデアル分解を，手計算でなくコンピュータ上で効率的に実行できれば，例えば素数 p が $\mathbb{Q}(\sqrt{d})$ の整数環において素元となる条件について簡単に予想を立てることもできる（具体的には例 3.3.4 で述べる）．他にも，上で述べた方程式が (x, y) について解を持つ条件を考察する際など，様々な問題において予想を立てるのに役立てることができる．

以上のように，整数論の研究の発展のために，高速な素イデアル分解のアルゴリズムを実現することが本論文の目的である．

2　研究方法

効率的なアルゴリズムを実現するため，抽象代数の視点から二次体の性質について考察を行った．また，考察により得られたアルゴリズムをプログラミング言語 C++ を用いて実装し，実際にいくつかの例に対しコンピュータ上で計算を行った．さらに，その結果から数学的な予想を立てることで，考案したアルゴリズムの有用性を確認した．理論においては，特に参考文献の［1］や［2］の内容を参考にして考察及び命題の証明を行った．

3　得られた結果・考察

この章では，d を平方因子を持たず，$0,1$ でもない整数とし，A を $\mathbb{Q}(\sqrt{d})$ の整数環とする．すなわち，$d\equiv 1 \pmod 4$ の時 $A=\mathbb{Z}\left[\frac{1+\sqrt{d}}{2}\right]$，$d\equiv 2,3 \pmod 4$ の時 $A=\mathbb{Z}[\sqrt{d}]$ である（これは［1］命題 8.5.1 より成り立つ）．以下，α を $d\equiv 1$ の時 $\frac{1+\sqrt{d}}{2}$，$d\equiv 2,3$ の時 $\sqrt{d}$ とする．

また，$\mathrm{C}(s+t\alpha)$ を $d\equiv 1$ なら $s+t\frac{1-\sqrt{d}}{2}$，$d\equiv 2,3$ なら $s-t\sqrt{d}$ とする．さらに $\mathrm{N}(x)=x\mathrm{C}(x)$ とし，これを x のノルムと呼ぶ．この時，$d\equiv 1$ なら $\mathrm{N}\left(s+t\frac{1+\sqrt{d}}{2}\right)=s^2+st+t^2\frac{1-d}{4}$，$d\equiv 2,3$ なら $\mathrm{N}\left(s+t\sqrt{d}\right)=s^2-t^2d$ となり，これらは整数である．

3.1　剰余

二次体における計算をする際，ある元 r の生成するイデアルによる類別を効率的に行わなければならないことが多くある．これは例えば $\mathbb{Z}$ においては，$x\in\mathbb{Z}$ にそれを r で割ったあまりを割り当てる，という操作により実現される．ここでは，二次体におけるそのような割り当ての方法を考える．

以下，$r=s+t\alpha(s,t\in\mathbb{Z})$ を零でない A の元とし，r の生成するイデアルによる剰余類のうち $x\in A$ の属するものを $\overline{x}$ と書く．さらに，u を $d\equiv 1$ の時 $\gcd\left(s,t\frac{1-d}{4}\right)$，$d\equiv 2,3$ の時 $\gcd(s,td)$ とし，$v=|\mathrm{N}(r)|/u$ とする．この u,v により，

$$R=\{a+b\alpha \mid a,b\in\mathbb{Z},0\le a<u,0\le b<v\}$$

とする．これが $A/\langle r\rangle$ の完全代表系を成すことを確かめ，$x\in A$ と属する剰余類が一致する R の元を効率的に求められることを示す．

まず初めに，前提となる次の命題を述べておく．これは［2］命題 1.10.17 の一部である．

命題 3.1.1（［2］の命題 1.10.17） $A/\langle r\rangle$ の位数は $|\mathrm{N}(r)|$ に等しい．

この命題より，R の位数は $A/\langle r\rangle$ の位数に一致している．

次の補題はさまざまな計算をする上で頻繁に用いる．

補題 3.1.2 (1) $\overline{n\alpha}=\overline{0}$ の整数解は v の倍数であるような n によって尽くされ，したがって $n=v$ が最小の正整数解である．
(2) $\overline{\mathrm{N}(r)}=\overline{0}$.
(3) $a+b\alpha\in R$ が $\overline{a+b\alpha}=\overline{0}$ を満たす $\Leftrightarrow a=b=0$.

証明 整数 x が素数 p で割れる回数を $\mathrm{ord}_p(x)$ と書く．（ただし $\mathrm{ord}_p(0)=\infty$ とする．）
(1) $\overline{n\alpha}=\overline{0}$ は $n\alpha/r\in A$ に同値であり，したがって $n\alpha\mathrm{C}(r)/\mathrm{N}(r)\in A$ に帰着される．$d\equiv 2,3 \pmod 4$ の時，$n\alpha\mathrm{C}(r)=-ntd+ns\sqrt{d}$ より，$1,\sqrt{d}$ の係数を見て，ns,ntd がともに $\mathrm{N}(r)$ の倍数であることが条件である．したがって，n が $\mathrm{N}(r)/\gcd(\mathrm{N}(r),s)$ と $\mathrm{N}(r)/\gcd(\mathrm{N}(r),td)$ の公倍数であれば良い．ここで，任意の素数 p に対して，

$$\begin{aligned}
&\mathrm{ord}_p\left(\mathrm{lcm}\left(\mathrm{N}(r)/\gcd(\mathrm{N}(r),s),\mathrm{N}(r)/\gcd(\mathrm{N}(r),td)\right)\right)\\
&=\mathrm{ord}_p(\mathrm{N}(r))-\min\left(\mathrm{ord}_p(s),\mathrm{ord}_p(td),\mathrm{ord}_p(\mathrm{N}(r))\right)\\
&=\mathrm{ord}_p(\mathrm{N}(r)/\gcd(s,td,\mathrm{N}(r)))\\
&=\mathrm{ord}_p(\mathrm{N}(r)/\gcd(s,td))
\end{aligned}$$

が成り立つ．なお，最後の行では $\mathrm{N}(r) = s^2 - t^2 d$ より $\gcd(s, td)$ が $\mathrm{N}(r)$ を割ることを使った．これによって， $\mathrm{lcm}(\mathrm{N}(r)/\gcd(\mathrm{N}(r), s), \mathrm{N}(r)/\gcd(\mathrm{N}(r), td)) = \mathrm{N}(r)/\gcd(s, td) = v$ である．よって， $\overline{n\sqrt{d}} = \overline{0}$ を整数 n が満たすことは， n が $v = \mathrm{N}(r)/\gcd(s, td)$ の倍数であることに同値．$d \equiv 1 \pmod 4$ の時も同様の計算により証明できる．

(2) $\mathrm{N}(r) = r\mathrm{C}(r)$ より成立．

(3) $a = b = 0 \Rightarrow \overline{a + b\alpha} = \overline{0}$ は明らかに成り立つので，逆を示す． $\overline{a + b\alpha} = \overline{0} \Leftrightarrow \exists m, n \in \mathbb{Z}\ s.t.\ a + b\alpha = (m + n\alpha)r$ である．$d \equiv 2, 3 \pmod 4$ の時， $(m + n\sqrt{d})r = ms + ntd + (mt + ns)\sqrt{d}$ より， $ms + ntd = a$ が整数解を持つことが必要．ゆえに a は $u = \gcd(s, td)$ の倍数でなければならない．R の定義より $a < u$ であるから $a = 0$ が必要であり，(1) と R の定義により $b = 0$ である．$d \equiv 1 \pmod 4$ の時も同様である． □

命題 3.1.3 R は $A/\langle r \rangle$ の完全代表系である．

証明 命題 3.1.1 より $|A/\langle r \rangle| = |R| = |\mathrm{N}(r)|$ であるから， R の元が $\langle r \rangle$ による剰余で全て相異なることを示せば良い．$z, w \in R$ を任意にとった時， $z - w = a + b\alpha\ (a, b \in \mathbb{Z}, 0 \le a < u, -v < b < v)$ とする（ $a < 0$ であったなら， z, w を入れ替えればこのような a, b がとれる）．$b \ge 0$ の時， $a + b\alpha \in R$ となるから，補題 3.1.2 (3) より $\overline{z - w} = \overline{0} \Leftrightarrow z - w = 0$ である．$b < 0$ の時，補題 3.1.2 (1) より $\overline{a + b\alpha} = \overline{a + (b + v)\alpha}$ である．$a + (b + v)\alpha \in R$ なので，補題 3.1.2 (3) より $\overline{a + (b + v)\alpha} = \overline{0} \Leftrightarrow a = 0, b + v = 0$ が言える．しかし， $-v < b$ であったからこれは成立し得ない．以上により， $z, w \in R$ に対し $\overline{z} = \overline{w} \Leftrightarrow z = w$ である． □

次の系は命題 3.1.3 から明らかである．

系 3.1.4 任意の $x \in A$ に対して， $\{z + x \mid z \in R\}$ は $A/\langle r \rangle$ の完全代表系である．

さて， R が $A/\langle r \rangle$ の完全代表系であることが確かめられたので，次に $a + b\alpha$ と同じ剰余類に属する R の元を効率的に見つけるアルゴリズムを考える．そのために，以下では特に $0 \le a < |\mathrm{N}(r)|, 0 \le b < v$ を満たすような元について議論する．これは， $|\mathrm{N}(r)|, v\alpha$ がともに r の倍元なので， a, b をそれぞれ $|\mathrm{N}(r)|, v$ で割ったあまりに置き換えても属する剰余類は変わらないからである．

補題 3.1.5 $\overline{a + b\alpha} = \overline{0}$ を満たす $0 \le b < v$ は，

- a が u の倍数の時ちょうど一つ存在する．
- a が u の倍数でない時存在しない．

証明 補題 3.1.2 (3) の証明を見れば， a が u の倍数でない時条件を満たす b が存在しないことがわかる．さて，任意の u の倍数 nu に対して，系 3.1.4 より $\{z + nu \mid z \in R\} = \{a + b\alpha \mid nu \le a < (n+1)u, 0 \le b < v\}$ は完全代表系．この集合の元のうち $\overline{a + b\alpha} = \overline{0}$ を満たす元は $u \mid a$ を満たすのだから， $a = nu$ を満たす．b の唯一性も明らか． □

以上を用いて計算を行うが，具体的な計算の際に必要となる基本的なアルゴリズムについて述べる．

まずユークリッドの互除法についてだが，これは整数 a, b に対して $\gcd(a, b)$ 及び $ax + by = \gcd(a, b)$ を満たす整数 x, y を求める方法である．以降では $\mathrm{Euclid}(a, b)$ を，それを満たす (x, y) を求める関数とする．

また，これを用いれば， $x \equiv a_1 \pmod{m_1}, x \equiv a_2 \pmod{m_2}$ である時， $x \bmod \mathrm{lcm}(m_1, m_2)$ の値を求めることができる（ただし，これを満たす x の存在に同値な $a_1 \equiv a_2 \pmod{\gcd(m_1, m_2)}$ を仮定する）．具体的には， $(p, q) \leftarrow \mathrm{Euclid}(m_1, m_2)$ として， $a_1 + \frac{(a_2 - a_1)m_1 p}{\gcd(m_1, m_2)} \bmod \mathrm{lcm}(m_1, m_2)$ を返せば良い．これは

$a_2 + \frac{(a_1-a_2)m_2q}{\gcd(m_1,m_2)}$ に等しいことから適切な値であることがわかる．この計算方法を，ここではガーナーの方法と呼ぶことにする（ガーナーのアルゴリズムは，正確にはより一般に n 個の法とそれぞれにおける x の値が与えられた時の計算アルゴリズムである）．なお， ユークリッドの互除法は $O(\log\min(|a|,|b|))$ で動作し，したがって gcd の計算やガーナーの方法も $O(\log\min(|m_1|,|m_2|))$ で動作する．

ここで，与えられた元と同じ剰余類に属する R の元を求めるために，まず，与えられた X に対して $\overline{X+Y\alpha}=\overline{0}$ なる Y を求めることを考える．

命題 3.1.6 X が u の倍数の時，$\overline{X+Y\alpha}=\overline{0}$ となる Y は $\bmod v$ で一意に定まり，その値は $O(\log|\mathrm{N}(r)|))$ で求められる．

証明 Y が $\bmod v$ で一意に定まることは，補題 3.1.5 から明らか．以下では $d\equiv 2,3\pmod 4$ の場合について議論する．補題 3.1.2 と同様にして，$(X+Y\sqrt{d})(s-t\sqrt{d})/\mathrm{N}(r)=(Xs-Ytd+(Ys-Xt)\sqrt{d})/\mathrm{N}(r)\in A$ が条件．したがって $Ytd\equiv Xs, Ys\equiv Xt\pmod{\mathrm{N}(r)}$ を Y について解けば良い．ここで，$\gcd(td,\mathrm{N}(r))\mid Xs$，$\gcd(s,\mathrm{N}(r))\mid Xt$ を示せれば，各式の両辺及び法の値をそれぞれ $\gcd(td,\mathrm{N}(r)),\gcd(s,\mathrm{N}(r))$ で割ることができる．X は u の倍数であったから， X を $u=\gcd(s,td)$ に置き換えてその成立を示す：

(i) $\gcd(td,\mathrm{N}(r))\mid s\cdot\gcd(s,td)$.

$s^2=\mathrm{N}(r)+t^2d$ より $\mathrm{ord}_p\left(s^2\right)\geq\min\left(\mathrm{ord}_p(\mathrm{N}(r)),\mathrm{ord}_p\left(t^2d\right)\right)\geq\min\left(\mathrm{ord}_p(\mathrm{N}(r)),\mathrm{ord}_p(td)\right)$ ゆえ，

$$\begin{aligned}\mathrm{ord}_p(s\cdot\gcd(s,td)) &= \min\left(\mathrm{ord}_p\left(s^2\right),\mathrm{ord}_p(s)+\mathrm{ord}_p(td)\right)\\ &\geq \min\left(\mathrm{ord}_p(\mathrm{N}(r)),\mathrm{ord}_p(td),\mathrm{ord}_p(s)+\mathrm{ord}_p(td)\right)\\ &= \min\left(\mathrm{ord}_p(\mathrm{N}(r)),\mathrm{ord}_p(td)\right)\end{aligned}$$

となり，確かに $\gcd(td,\mathrm{N}(r))\mid s\cdot\gcd(s,td)$ が成り立つ．

(ii) $\gcd(s,\mathrm{N}(r))\mid t\cdot\gcd(s,td)$.

$t^2d=s^2-\mathrm{N}(r)$ より， $\mathrm{ord}_p\left(t^2d\right)\geq\min\left(\mathrm{ord}_p\left(s^2\right),\mathrm{ord}_p(\mathrm{N}(r))\right)$ が成り立つので，

$$\begin{aligned}\mathrm{ord}_p(t\cdot\gcd(s,td)) &= \min\left(\mathrm{ord}_p(st),\mathrm{ord}_p\left(t^2d\right)\right)\\ &\geq \min\left(\mathrm{ord}_p(st),\mathrm{ord}_p\left(s^2\right),\mathrm{ord}_p(\mathrm{N}(r))\right)\\ &\geq \min\left(\mathrm{ord}_p(s),\mathrm{ord}_p(\mathrm{N}(r))\right)\end{aligned}$$

となり， $\gcd(s,\mathrm{N}(r))\mid t\cdot\gcd(s,td)$ がわかる．

以上により，解くべきは

$$\begin{aligned}\frac{td}{\gcd(td,\mathrm{N}(r))}Y &\equiv \frac{Xs}{\gcd(td,\mathrm{N}(r))}\left(\bmod\frac{\mathrm{N}(r)}{\gcd(td,\mathrm{N}(r))}\right),\\ \frac{s}{\gcd(s,\mathrm{N}(r))}Y &\equiv \frac{Xt}{\gcd(s,\mathrm{N}(r))}\left(\bmod\frac{\mathrm{N}(r)}{\gcd(s,\mathrm{N}(r))}\right)\end{aligned}$$

に帰着される．さて， $\mathbb{Z}/n\mathbb{Z}$ における $a+n\mathbb{Z}$ $(a\in\mathbb{Z})$ の乗法逆元は，a と n が互いに素ならば $ax+ny=1$ をユークリッドの互除法で解くことで求められる．今回はこれを Y の係数と法の値に適用することで，Y の $\mathrm{mod}\frac{\mathrm{N}(r)}{\gcd(td,\mathrm{N}(r))},\mathrm{mod}\frac{\mathrm{N}(r)}{\gcd(s,\mathrm{N}(r))}$ での値を求められる．したがって，あとはガーナーの方法を適用すれば， Y の $\mathrm{mod}\left(\mathrm{lcm}\left(\frac{\mathrm{N}(r)}{\gcd(td,\mathrm{N}(r))},\frac{\mathrm{N}(r)}{\gcd(s,\mathrm{N}(r))}\right)\right)$ での値が求められる．補題 3.1.2 の証明を見れば，$\mathrm{lcm}\left(\frac{\mathrm{N}(r)}{\gcd(td,\mathrm{N}(r))},\frac{\mathrm{N}(r)}{\gcd(s,\mathrm{N}(r))}\right)=v$ であったから，結局， Y が $\bmod v$ で求められる．計算量は， gcd の計算などに現れる値が全て $|\mathrm{N}(r)|$ で抑えられるので，全体で $O(\log|\mathrm{N}(r)|)$ である．

$d \equiv 1$ の時も同様の計算を行えば，$D = \frac{1-d}{4}$ として

$$\frac{tD}{\gcd(tD, \mathrm{N}(r))} Y \equiv \frac{-X(s+t)}{\gcd(tD, \mathrm{N}(r))} \left(\mathrm{mod} \frac{\mathrm{N}(r)}{\gcd(tD, \mathrm{N}(r))} \right),$$
$$\frac{s}{\gcd(s, \mathrm{N}(r))} Y \equiv \frac{Xt}{\gcd(s, \mathrm{N}(r))} \left(\mathrm{mod} \frac{\mathrm{N}(r)}{\gcd(s, \mathrm{N}(r))} \right).$$

に帰着され，$O(\log |\mathrm{N}(r)|)$ で計算が可能である． □

この補題を用いれば，剰余を求める関数を書くことができる．具体的には次のようにすれば良い．

Remainder$(a + b\alpha, r)$：
Input: $a + b\alpha, r \in A$.
Output: $x \in R$ *s.t.* $a + b\alpha = x \pmod{\langle r \rangle}$.

- a, b をそれぞれ $|\mathrm{N}(r)|, v$ で割ったあまりに置き換える．
- a 以下の最大の u の倍数を X とし，$\overline{X + Y\alpha} = \overline{0}$ を満たす最小の正整数 Y を求める．
- $(a - X) + (b - Y)\alpha$ または $(a - X) + (b - Y + v)\alpha$ は R の元であるので，これを返す．

例 3.1.7 Remainder 関数の実行例を挙げておく．

- $d = -5$ の時，$\mathrm{Remainder}(6, 1 + \sqrt{-5}) = 0$ である（$6 = (1 + \sqrt{-5})(1 - \sqrt{-5})$ が成り立つため）．
- $d = -2$ の時，$\mathrm{Remainder}(31 + 41\sqrt{-2}, 59 + 26\sqrt{-2}) = 355\sqrt{-2}$ である．この例においては，$u = \gcd(59, -2 \cdot 26) = 1$ であるから，必ず剰余は $b\sqrt{-2}$ という形になる．

```
347    using std::cin;
348    using std::cout;
349    using std::endl;
350
351    long long d;
352    int p,q,r,s;
353
354    int main(){
355        cin >> d;
356        using A=ring_of_integer<d>;
357        cin >> p >> q;
358        cin >> r >> s;
359        A::elem x(p,q),y(r,s);
360        A::elem res=Remainder(x,y);
361        //Remainder(x,y)の結果を出力する.
362        cout << res << endl;
363        //res-xがyで割り切れることを確認する. 割り切れるなら1を出力する.
364        cout << (res-x).is_divided_by(y) << endl;
365    }
```

問題　出力　デバッグ コンソール　ターミナル

```
m.i.@logx-mac KdEi-quadratic-field % g++ quadratic_field.cpp
m.i.@logx-mac KdEi-quadratic-field % ./a.out
-2
31 41
59 26
355\sqrt{-2}
1
```

$\mathrm{Remainder}(31 + 41\sqrt{-2}, 59 + 26\sqrt{-2})$ を実行した結果．ターミナルの下 2 行が出力である．

3.2 イデアルの生成元

一般に，デデキント環においてイデアルは高々 2 つの元によって生成されることが言える．そこでここでは， A のイデアルが与えられた時，それを生成する 2 つの A の元を探すアルゴリズムを考える．まず次の補題を述べておく．これは［3］9 章演習問題 7 の命題である．

補題 3.2.1（［3］9 章の演習問題 7） デデキント環 A において，任意の $r \in A \setminus \{0\}$ に対して $A/\langle r\rangle$ の任意のイデアルは単項生成であり，したがって A のイデアルは高々 2 元により生成される．

命題 3.2.2 次のアルゴリズムにより， $\langle F\rangle$ の生成元を計算量 $O((|\mathrm{N}(x)|^2 + |F|\,|\mathrm{N}(x)|)\log|\mathrm{N}(x)|)$ で求めることができる．ただし F の，零でないノルムの絶対値が最小の元を x とする．

Generator(F)：
Input: 集合 $F \subset A$.
Output: $\langle F\rangle = \langle x, y\rangle$ なる $x, y \in A$.

1. $F = \{0\}$ の時： $(0, 0)$ を return する．
2. $x \leftarrow F$ の零でないノルム最小の元， $I \leftarrow \{0\}$ ， $q \leftarrow \{0\}$ とする．
3. q が空でない間，次を実行する：
   - w を q のいずれかの元として q から w を除く．
   - 各 $z \in \{\mathrm{Remainder}(w + t, x) \mid t \in F\} \cup \{\mathrm{Remainder}(w + t\alpha, x) \mid t \in F\} \setminus I$ に対して， $I \leftarrow I \cup \{z\}, q \leftarrow q \cup \{z\}$ とする．
4. $X \leftarrow \{0\}$ とし，各 $t \in I$ に対して次を実行する：
   - $t \in X$ である時：次の I の元へ．
   - $J \leftarrow \{0\}, q \leftarrow \{0\}$ として，3. と同様のアルゴリズムを実行する．ただし F を $\{t\}$ に， I を J に置き換える．
   - $|I| = |J|$ である時： (x, t) を return する．
   - $X \leftarrow X \cup J$ として次の I の元へ．

証明 まず，3. の操作により， I に $\langle F\rangle/\langle x\rangle$ の元が全て入ることを示す（なお， $A/\langle x\rangle$ の元は，3.1 節で r に対して定義した R と同様の元により代表させる）．$\langle F\rangle$ の元は $\sum_{t\in F} t(m_t + n_t\alpha)$ $(m_t, n_t \in \mathbb{Z})$ という形でかける． $\overline{|\mathrm{N}(x)|\,t} = \overline{|\mathrm{N}(x)|\,t\alpha} = \overline{0}$ であることにも気をつければ， $\langle F\rangle/\langle x\rangle$ の元は， $F' \stackrel{\mathrm{def}}{=} F \cup \{t\alpha \mid t \in F\}$ なる集合の有限和に対して Remainder 関数を適用したもの全体として表せる．したがって， 0 から開始して $\langle F\rangle/\langle x\rangle$ の元であることが確定した元から順に，その元に t ないし $t\alpha$ $(t \in F)$ を加算したものをさらに I に加える，という操作を繰り返せば良い．この部分は，最終的な I の各要素 w に対してちょうど一度ずつ $t, t\alpha$ $(t \in F)$ を加えた元を見るので， Remainder 関数の計算に $O(\log|\mathrm{N}(x)|)$ かかることと合わせて $O(|F|\,|I|\log|\mathrm{N}(x)|)$ で動作する．命題 3.1.1 から $|I| \le |\mathrm{N}(x)|$ ゆえ，これは $O(|F|\,|\mathrm{N}(x)|\log|\mathrm{N}(x)|)$ である．

次に 4. であるが，各 $t \in I$ に対して $\langle t\rangle/\langle x\rangle$ の元を列挙して J とし，それが I と一致しているか判定している．J に $\langle t\rangle/\langle x\rangle$ の元が全て入ることの証明は，先ほどの $\langle F\rangle/\langle x\rangle$ に対する証明において F を $\{t\}$ に置き換えれば良い．さて， $t \in X$ であることは既にみた他の元によって生成されたイデアルのいずれかに含まれる，という条件に同値である．この条件を確認している，つまりまだこの関数の実行が終

わっていないならば，そのイデアルは $I = \langle F\rangle/\langle x\rangle$ に等しくなかったということである．$\langle t\rangle/\langle x\rangle$ はそのイデアルに含まれるのだから，これも $\langle F\rangle/\langle x\rangle$ に一致し得ない．したがって，$t \in X$ を満たす t を調べずとも，$\langle F\rangle/\langle x\rangle$ の生成元として適切な候補は全て調べられており，補題 3.2.1 から必ず $I = J$ を満たすような t が見つかる．後半部分の計算量については，I の各元に対して $O(|J|\log|\mathrm{N}(x)|)$ の計算を行っているため，$O(\sum|J|\log|\mathrm{N}(x)|) = O(|I|\max\{|J|\}\log\mathrm{N}(x))$ である．$|J| \le |I| \le |\mathrm{N}(x)|$ より，これは $O(|\mathrm{N}(x)|^2\log|\mathrm{N}(x)|)$ である． □

さて，このアルゴリズムと同様の方法を用いて実現できる次のアルゴリズムを述べておく．計算量は $O(|F|\,|\mathrm{N}(r)|\log|\mathrm{N}(r)|)$ である（ただし r は零でない F のノルム最小の元）．

Contains(F, x)：
Input: 集合 $F \subset A$，$x \in A$.
Output: $x \in \langle F\rangle$ の真偽値.

1. $F = \{0\}$ の時，$x = 0$ の真偽値を return する．
2. $r \leftarrow F$ の零でないノルム最小の元として，$I \leftarrow \langle F\rangle/\langle r\rangle$ とする（Generator 関数と同様にする）．
3. Remainder$(x, r) \in I$ の真偽値を return する．

例 3.2.3 Generator 関数及び Contains 関数を実際にいくつか実行した結果を示しておく．

- $d = -2$ の時，Generator$(2, 3, 4)$ は $(2, 1)$ である．$\langle 2, 3, 4\rangle = \langle 2, 1\rangle = \langle 1\rangle$ より正しい．
- $d = -5$ の時，Generator$(36, 6 + 18\sqrt{-5}, 6 + 12\sqrt{-5}, -29 + 5\sqrt{-5}) = (6 + 12\sqrt{-5}, 1 + 11\sqrt{-5})$ である．
- $d = -5$ の時，Contains$(\{1 + \sqrt{-5}\}, 6)$ は true．$6 = (1 + \sqrt{-5})(1 - \sqrt{-5})$ より明らか．
- $d = -5$ の時，Contains$(\{1 + \sqrt{-5}\}, 5)$ は false.
- $d = 3$ の時，Contains$(\{1 + \sqrt{3}, 2\}, 7 + 9\sqrt{3})$ は true．実際，$7 + 9\sqrt{3} = 7(1 + \sqrt{3}) + 2\sqrt{3}$ である．

3.3 素イデアル分解

この節で，目標であった素イデアル分解のアルゴリズムを述べる．そのために，計算量解析の際に必要となる次の命題を述べておく．これは［2］命題 1.10.15 から得られる．

命題 3.3.1（［2］の命題 1.10.15） I, J を A の零でないイデアルとした時，$|A/I|\,|A/J| = |A/IJ|$ が成り立つ．

なお，素イデアル $\mathfrak{p}$ に対しては明らかに $|A/\mathfrak{p}| > 1$ であることに注意しておく．

命題 3.3.2 次のアルゴリズムは正しく動作し，計算量は $O(|\mathrm{N}(g_1)|^2\log^2|\mathrm{N}(g_1)|)$ である．

PrimeFactorize(g_1, g_2)：
Input: イデアル I の生成元 g_1, g_2.
Output: I を素イデアル分解した結果を $I = \prod_i \mathfrak{p}_i^{a_i}$ として，$(\mathfrak{p}_i, a_i)$ なる素イデアルと整数の組の列.

1. $X \leftarrow \emptyset, Y \leftarrow \emptyset$（それぞれ，他の真のイデアルに含まれていた元，極大イデアルの生成元の候補を保持する集合.）

2. $A/\langle g_1\rangle$ の代表系の各要素 t に対して次を実行する：
   - $t \in X$ の時：次の元へ.
   - $I \leftarrow \langle t\rangle/\langle g_1\rangle$ とする（Generator 関数と同様にすれば良い）.
   - $|I| = |A/\langle g_1\rangle|$ の時：次の元へ.
   - $X \leftarrow X \cup I, Y \leftarrow (Y \setminus I) \cup \{t\}$.
3. $S \leftarrow \emptyset$
4. 各 $t \in Y$ に対して次を実行する：
   - $n \leftarrow 0, (p, q) \leftarrow (g_1, t)$
   - $\langle p, q\rangle$ が g_1, g_2 をともに含む間（Contains 関数により判定する），次を行う：
     - $n \leftarrow n + 1, (p, q) \leftarrow \mathrm{Generator}(\{pg_1, pg_2, qg_1, qg_2\})$.
   - $n \neq 0$ の時， $S \leftarrow S \cup \{(\langle g_1, t\rangle, n)\}$ とする.
5. S を return する.

証明 まず，前半で列挙した Y の元が $A/\langle g_1\rangle$ の極大イデアルの生成元全体であることを示す．このために，$A/\langle g_1\rangle$ の代表系の元に対し，2. でみた順に $t_1, t_2, \ldots, t_{|\mathrm{N}(g_1)|}$ と番号をふる.

$t_i \in X$ であったなら，ある $j < i$ が存在して $\langle t_i\rangle/\langle g_1\rangle \subset \langle t_j\rangle/\langle g_1\rangle$ となるのだから t_i は極大イデアルを生成し得ず，また，$\langle t_i\rangle/\langle g_1\rangle$ が全体に一致する場合も明らかに t_i は極大イデアルを生成しない．さらに，Y から I の元を除く操作があるが，I に含まれる元によって生成されたイデアルは明らかに I に含まれるのだから，t_i 以外の I の元は極大イデアルを生成しないか，I を生成するかの何れかである．したがって，Y に加えなかった元は極大イデアルを生成し得ず，さらに，Y から除く操作を行った元も，極大イデアルを生成しないか t_i と同じ極大イデアルを生成するかのどちらかである．t_i はあとから Y に追加されるのだから，結局，Y には任意の極大イデアルの生成元が一つ以上含まれることがわかる.

また，極大イデアルを生成しない元 t_i について，t_j $(j < i)$ が t_i を含む真のイデアルを生成するなら，そのような最小の j により $t_i \in X$ となり t_i は Y に追加され得ない．このような j が存在しないならば，t_j $(j > i)$ が t_i を含む真のイデアルを生成する，という条件を満たす最小の j をとれば，t_j の生成したイデアルの元を除く操作により t_i は Y から除かれる．さらに，同じ極大イデアルを生成する複数の元は，同様に Y から I の元を除く，という操作により同時に Y の元とはなり得ない.

以上により，Y は $A/\langle g_1\rangle$ の各極大イデアルの生成元を一つずつ含む．なお，ここまでの計算量は $O(|\mathrm{N}(g_1)|^2 \log |\mathrm{N}(g_1)|)$ である.

さて，後半部分についてだが，素イデアルの冪 $\langle g_1, t\rangle^n$ が $\langle g_1, g_2\rangle$ を含むような最大の n を選ぶ操作をしているだけであり，適切に動作することは明らかである（素イデアル分解の一意性から，これは有限回の操作で終了する）．Contains 関数及び Generator 関数による計算が何度も行われるが，これを呼び出す回数は $O(\log |\mathrm{N}(g_1)|)$ で抑えられることを示す．以下，Y の i 番目の元 t に対して $\mathfrak{p}_i = \langle g_1, t\rangle, a_i = n$ とし，さらに $\langle g_1\rangle = \prod \mathfrak{p}_i^{b_i}$ とする．$\langle g_1\rangle \subset \langle g_1, g_2\rangle$ より，$a_i \leq b_i$ である．さらに，$\mathfrak{p}_i$ は $\langle g_1\rangle$ を含むのだから，$b_i > 0$ ．さて，関数呼び出しの回数は $\sum_i (a_i + 1) \leq \sum_i (b_i + 1) \leq \sum_i 2b_i$ である．$|A/\mathfrak{p}_i| = p_i$ とすれば，命題 3.3.1 から $\prod p_i^{b_i} = |A/\langle g_1\rangle| = |\mathrm{N}(g_1)|$ ．$p_i \geq 2$ に気をつければ，$\sum_i b \leq \log_2 |\mathrm{N}(g_1)|$ である．よって，$\sum_i (a_i + 1) \leq 2\sum_i b_i \leq 2\log_2 |\mathrm{N}(g_1)| = O(\log |\mathrm{N}(g_1)|)$ が成り立つ．Contains 関数と Generator 関数の計算量は，与える生成元の個数が定数の時 $O(|\mathrm{N}(g_1)| \log |\mathrm{N}(g_1)|)$ 及び $O(|\mathrm{N}(g_1)|^2 \log |\mathrm{N}(g_1)|)$ であったから，4. は全体として $O(|\mathrm{N}(g_1)|^2 \log^2 |\mathrm{N}(g_1)|)$ で動作する.

以上から，この素イデアル分解のアルゴリズムは $O(|\mathrm{N}(g_1)|^2 \log^2 |\mathrm{N}(g_1)|)$ で動作する． □

例 3.3.3 素イデアル分解を実行した結果をいくつか以下に示す.

- $d=2$ の時, $\langle 2\rangle=\langle 2,\sqrt{2}\rangle^2$ である. 実際, 右辺は $\langle\sqrt{2}\rangle^2=\langle 2\rangle$ に等しい.
- $d=-5$ の時, $\langle 6\rangle=\langle 6,1+2\sqrt{-5}\rangle^1\cdot\langle 6,1+3\sqrt{-5}\rangle^2\cdot\langle 6,1+4\sqrt{-5}\rangle^1$ である. 実際, 右辺を Generator 関数などを用いて計算すると $\langle 12+24\sqrt{-5},6\sqrt{-5}\rangle$ に等しく, これは $\langle 6\rangle$ に一致している.

```
347  using std::cin;
348  using std::cout;
349  using std::endl;
350
351  long long d=-5;
352
353  int main(){
354      using A=ring_of_integer<d>;
355      A::ideal I({6}),J({1});
356      for(auto [P,n]:I.PrimeFactorize()){
357          cout << '(' << P.gen[0] << ',' << P.gen[1] << ")^" << n << endl;
358          while(n--)J=P*J;
359      }
360      cout << "J=(" << J.gen[0] << ',' << J.gen[1] << ')' << endl;
361      cout << "I=J is " << (I==J ? "true":"false") << endl;
362  }
```

問題　出力　デバッグ コンソール　ターミナル

```
m.i.@logx-mac KdEi-quadratic-field % g++ quadratic_field.cpp
m.i.@logx-mac KdEi-quadratic-field % ./a.out
(6,1+2\sqrt{-5})^1
(6,1+3\sqrt{-5})^2
(6,1+4\sqrt{-5})^1
J=(12+24\sqrt{-5},6\sqrt{-5})
I=J is true
```

$\langle 6\rangle$ の素イデアル分解を実行した様子.

例 3.3.4 素イデアル分解を利用して, p の生成するイデアルが $\mathbb{Q}(\sqrt{d})$ の整数環で分解される条件について予想を立てることができる.

- $d=-1$ として, 各素数 p に対して $\langle p\rangle$ の素イデアル分解を行い, 結果として $\langle p\rangle$ が分解されなかったような p を小さい方から出力すると, $p=3,7,11,19,23\ldots$ となる. $\mathbb{Z}[\sqrt{-1}]$ は一意分解整域であるので, $\langle p\rangle$ が分解されることは $p=(x+y\sqrt{-1})(x-y\sqrt{-1})$ となる整数 (x,y) が存在することと同値であることが示せる. 素数 p が $p=x^2+y^2$ と表せない条件が $p\equiv 3\pmod 4$ であることは有名であるが, この実行結果によりその事実を確認することができる.
- $d=-5$ として, 同様の操作を $2\le p\le 100$ なる素数に対して行うと, 出力されるのは $p=11,13,17,19,31,37,53,59,71,73,79,97$ となる. これを観察すれば, $p\equiv 11,13,17,19\pmod{20}$ が「分解されない条件」だろう, という予想を立てることができる. 実際,「$\mathbb{Z}[\sqrt{-5}]/\langle p\rangle$ が整域である」と言い換えて条件を整理すれば, ルジャンドル記号を用いて $\left(\frac{-5}{p}\right)=-1$ に帰着され, 平方剰余の相互法則などを用いて計算すれば, これは $p\equiv 11,13,17,19\pmod{20}$ に同値であることが示せる.

この節で紹介した関数 Remainder, Generator, Contains および PrimeFactorize を実装したコードは, GitHub 上で公開しており, 誰でも利用可能である. リンクはこちら:
`https://github.com/mitsu-a/JSEC_algorithms`

4　結論

本研究を通して，初等的な計算を主に用いた方法で二次体の整数環における様々なアルゴリズムを作ることができた．さらに，それを用いることで，容易に解についての予想が立てられる問題があること（例 3.3.4）を示せた．今回示した問題は簡単に解けるものだったが，他の二次体における計算に帰着される問題を解く際など，予想を立てたり，結果が正しいことの確認に用いたりすることが期待できる．

一方で，Generator 関数や PrimeFactorize 関数において，計算量が $O(|\mathrm{N}(x)|^{2+\epsilon})$ 程度（ x は適当な A の元）になったのは今後の研究課題である．上で述べたような， $p \leq 100$ に対して計算をするような場合には問題ないが，より大きな値に対して計算をしようとするとかなりの時間がかかってしまう．例えば $s \in \mathbb{Z}$ に対してノルムは s^2 であることから， $O(|\mathrm{N}(s)|^{2+\epsilon}) = O(s^{4+\epsilon})$ であり，大きな s に対して高速に計算ができるとは言えない．これを高速化し，例えば $O(|\mathrm{N}(x)|^{1+\epsilon})$ 程度の計算量を目指した研究を行おうと考えている．

また，今回は用いていないものの，剰余の計算などにグレブナー基底を使う手法が知られている．二次体に限らない一般の代数体における計算を行う上では，多項式環の剰余環における計算が必須になると考えられる．そこで，今後はグレブナー基底をはじめとする様々な概念についても学習し，実装に活用していきたいと考えている．

謝辞

本研究を進めるにあたり，多くの方にお世話になりました．私の参加する角川ドワンゴ学園研究部では，東京理科大学理学部第一部応用数学科の石原侑樹助教にアドバイザーを担当して頂き，論文の書き方の指導などの様々なサポートをして頂きました．本当にありがとうございました．また，研究部の数理科学グループのメンバーやアドバイザーの方には，毎月のメンタリングにおいて多くの刺激やアドバイスを頂きました．心から感謝致します．さらに，学校法人角川ドワンゴ学園及びその研究部の運営に携わっている方々には，このような研究の機会を提供して頂いたことに感謝を申し上げます．

参考文献

この研究を行うに当たって，以下の文献を参考にした．

［1］ 雪江明彦，『整数論 1 初等整数論から p 進数へ』，日本評論社，2013

［2］ 雪江明彦，『整数論 2 代数的整数論の基礎』，日本評論社，2013

［3］ M.F.Atiyah and I.G.MacDonald （新妻弘訳），『可換代数入門』，共立出版，2006