

## 2.5 既約剰余類

合同類の積のところでも述べたように、 $\mathbb{Z}_m - \{0\}$  は、 $m$  が素数でなければ、積に関して群にならない。なぜなら、 $m = pq$  となっているなら、 $pq \equiv 0 \pmod{m}$  であるから、 $p, q \in \mathbb{Z}_m$  とみなした時に  $pq = 0 \in \mathbb{Z}_m$  となってしまい、 $p, q$  は  $\mathbb{Z}_m - \{0\}$  の中では積に関する逆元をもたないからである。(  $p^{-1}$  があれば  $0 = p^{-1}pq = q$  となって  $q \in \mathbb{Z}_m - \{0\}$  に矛盾する。)

しかし、 $\{1, 2, \dots, m-1\} = \mathbb{Z}_m - \{0\}$  の中には  $m$  と互いに素な数も存在する。今、 $p \in \{1, 2, \dots, m-1\}$  を  $\text{GCD}(p, m) = 1$  となる数とすると、定理 2.2 より、ある整数  $k_1, k_2$  があって  $k_1p + k_2m = 1$  となる。 $k_1p = -k_2m + 1$  なので、

$$k_1p \equiv 1 \pmod{m}$$

となる。これは、 $k_1$  を含む  $\mathbb{Z}_m$  の剰余類  $[k_1]$  が  $\mathbb{Z}_m - \{0\}$  において、積に関する  $p$  の逆元であることを示している。

**定義 2.10**  $m \geq 2$  を自然数とする。

- (1)  $p \in \{1, \dots, m-1\}$  が  $m$  と互いに素であるなら、 $p$  を含む  $\mathbb{Z}_m$  の剰余類を **既約剰余類** という。
- (2)  $\mathbb{Z}_m$  における既約剰余類全体の集合を  $\mathbb{Z}_m^*$  で表す。 $m$  が素数なら  $\mathbb{Z}_m^* = \mathbb{Z}_m - \{0\}$  であるが、一般にはそうなるとは限らない。
- (3)  $\mathbb{Z}_m^*$  の元は積に関して逆元を持っているので、この集合は積に関して群をなす。これを **既約剰余類群** という。
- (4)  $\varphi(m) = |\mathbb{Z}_m^*|$  とおき、これを **オイラー関数** という。 $\varphi(1) = 1$  と決める。

**注意 1.** オイラー関数  $\varphi(m)$  は、 $\{1, \dots, m-1\}$  の中で  $m$  と互いに素であるものの個数を表す。例えば、 $p$  が素数ならば  $\varphi(p) = p-1$  である。また、

$$\varphi(4) = 2, \quad \varphi(6) = 2, \quad \varphi(8) = 4, \quad \varphi(9) = 6, \quad \varphi(10) = 4$$

など。

**注意 2.** 一般に、既約剰余類群は巡回群とは限らない。例えば、 $\mathbb{Z}_8^* = \{1, 3, 5, 7\}$  であるが、 $\mathbb{Z}_8^*$  において、

$$3^2 = 1 \quad 5^2 = 1 \quad 7^2 = 1$$

となって、単位元以外の全ての元の位数が 2 であり、どの元も  $\mathbb{Z}_8^*$  全体を生成しない。

定理 1.39 より次の定理が成り立つ。これは、フェルマーの小定理の一般形であり、オイラーによって証明された。

定理 2.11 (一般化されたフェルマーの小定理)  $m \geq 2$  を自然数とする。

任意の  $a \in \mathbb{Z}_m^*$  に対して

$$a^{\varphi(m)} = 1$$

が成り立つ。言い換えると、 $a \in \mathbb{Z}$  が  $m$  と互いに素ならば、

$$a^{\varphi(m)} \equiv 1 \pmod{m}$$

である。

## 2.6 拡張ユークリッド・アルゴリズム

前セクションで見たように、 $m \geq 2$  を自然数とすると、 $\mathbb{Z}_m^*$  の任意の元  $e$  は積に関する逆元  $e^{-1}$  を持つ。(なお、ここで  $e$  は積に関する単位元ではなく、単なる  $\mathbb{Z}_m^*$  の元の意味で使っているので注意。)

それでは具体的に、 $m$  と互いに素な  $e \in \{1, 2, \dots, m-1\}$  が与えられた時、その  $\mathbb{Z}_m^*$  における逆元  $e^{-1} \in \{1, 2, \dots, m-1\}$  はどのように計算したら良いのだろうか? 実はこの計算は、RSA 暗号において、その根幹をなす重要な部分となる。

$m$  と  $e$  は互いに素なので、定理 2.2 より、ある整数  $k_1, k_2$  があって、 $k_1e + k_2m = 1$  となる。 $k_1e = -k_2m + 1$  なので、 $k_1e \equiv 1 \pmod{m}$  であり、これは、 $k_1$  を含む剰余類が  $e$  の逆元であることを示している。

従って、 $e \in \mathbb{Z}_m^*$  の逆元を求める問題は、

- $k_1e + k_2m = 1$  を満たす  $k_1$  を求める問題である。

と言える。 $k_1$  が求まれば自動的に  $k_2$  も決まってしまうので、これは、

- $k_1e + k_2m = 1$  を満たす  $(k_1, k_2)$  を求める問題。

とすることができる。

この  $(k_1, k_2)$  は、実はユークリッドの互除法を利用することにより求めることができる。この方法を **拡張ユークリッド・アルゴリズム** という。それは、以下のようにして行う。

$e \in \{1, 2, \dots, m-1\}$  であるので  $e < m$  である。 $GCD(m, e)$  を求めるユークリッドの互除法を行ってみる。 $GCD(m, e) = 1$  なので、これは最後の余りが 1 になった時点で終了する。

$$\begin{aligned} m &= q_1e + r_1 \\ e &= q_2r_1 + r_2 \\ r_1 &= q_3r_2 + r_3 \\ r_2 &= q_4r_3 + r_4 \\ &\dots \quad \dots \\ r_{s-2} &= q_sr_{s-1} + r_s \\ r_{s-1} &= q_{s+1}r_s + r_{s+1} \\ r_s &= q_{s+2}r_{s+1} + 1 \end{aligned}$$

となったとする。

- $m = q_1e + r_1$  より  $r_1 = m - q_1e$  である。すなわち、最初の余り  $r_1$  は  $m$  と  $e$  の 1 次結合で表されている。
- $e = q_2r_1 + r_2$  より  $r_2 = e - q_2r_1$  である。すなわち、 $r_2$  は  $e$  と  $r_1$  の 1 次結合で表されている。しかし、最初の式から、 $r_1$  は  $m$  と  $e$  の 1 次結合で表されているので、結局、 $r_2$  は  $m$  と  $e$  の 1 次結合で表される。
- $r_1 = q_3r_2 + r_3$  より  $r_3 = r_1 - q_3r_2$  である。すなわち、 $r_3$  は  $r_1$  と  $r_2$  の 1 次結合で表されている。しかし、 $r_1$  と  $r_2$  は共に  $m$  と  $e$  の 1 次結合で表されているので、結局、 $r_3$  は  $m$  と  $e$  の 1 次結合で表される。
- このプロセスを繰り返すことにより、全ての余り  $r_i$  は、 $m$  と  $e$  の 1 次結合で表されることになる。
- 最後の余りは 1 なので、最終的に、1 が  $m$  と  $e$  の 1 次結合で表されることになる。これにより、 $k_1e + k_2m = 1$  という式に到達する。

この方法を見るとわかるように、基本的にこれはユークリッド互除法であり、アルゴリズムの計算量はユークリッド互除法とほとんど同じである。すなわち、200 桁程度の  $m$  や  $e$  に対しては、現在の計算機の能力からすれば、極めて短時間で計算可能である。

例： 具体例で見てみよう。

$$36a + 47b = 1$$

を満たす整数  $a, b$  を求めることを考える。

47 は素数なので、当然  $GCD(36, 47) = 1$  であり、定理 2.3 より、求める整数解  $a, b$  は存在する。

ユークリッド互除法により、36, 47 の最大公約数 1 を求めるプロセスを実行してみると、

$$\begin{aligned} 47 &= 36 + 11 \\ 36 &= 3 \cdot 11 + 3 \\ 11 &= 3 \cdot 3 + 2 \\ 3 &= 2 + 1 \end{aligned}$$

となる。

拡張ユークリッド・アルゴリズムは、ユークリッド互除法を実行しながら、この余りの数 11, 3, 2, 1 を 36 と 47 の 1 次結合で表して行く、というプロセスである。

具体的には、

$$\begin{aligned} 11 &= -36 + 47 \\ 3 &= 36 - 3 \cdot 11 = 36 - 3 \cdot (-36 + 47) = 4 \cdot 36 - 3 \cdot 47 \\ 2 &= 11 - 3 \cdot 3 = (-36 + 47) - 3 \cdot (4 \cdot 36 - 3 \cdot 47) = -13 \cdot 36 + 10 \cdot 47 \\ 1 &= 3 - 2 = (4 \cdot 36 - 3 \cdot 47) - (-13 \cdot 36 + 10 \cdot 47) = 17 \cdot 36 - 13 \cdot 47 \end{aligned}$$

となり,  $a = 17, b = -13$  が得られた。

ちなみに, この  $17 \cdot 36 - 13 \cdot 47 = 1$  より,

$$36 \cdot 17 \equiv 1 \pmod{47}$$

がわかる。すなわち,  $\mathbb{Z}_{47}^*$  において, 36 の積に関する逆元は 17 である。

**演習問題 2.3** 互いに素である 2 つの自然数  $m, n$  を入力すると,  $\mathbb{Z}_m^*$  における  $n$  の逆元  $n^{-1}$  を出力するプログラムを作れ。可能なものは BigInteger クラスを用いていくらでも大きい自然数に対応可能なものを作れ。