アルゴリズム論特論（特講）　(塩田)

2008 年 4 月 24 日のレジュメ

今回紹介する系 4 とアルゴリズム 9 は、暗号理論、整数論、計算代数において極めて重要な役割を果たす。

1　$ax + by$ の形で表せる数

定理 1　$a, b \in \mathbf{Z}$ を与えたとき、$ax + by$ $(x, y \in \mathbf{Z})$ の形で表すことのできる整数全体の集合

$$\{\, ax + by \,|\, x, y \in \mathbf{Z} \,\}$$

は、$d = \gcd(a, b)$ の倍数全体の集合 $d\mathbf{Z} = \{\, dk \,|\, k \in \mathbf{Z} \,\}$ に一致する。

系 2　$a, b \in \mathbf{Z}$ が互いに素 $(\gcd(a, b) = 1)$ であれば、任意の整数を $ax + by$ $(x, y \in \mathbf{Z})$ の形で表すことができる。

系 3　$a, b \in \mathbf{Z}$ を与えたとき、$ax + by = \gcd(a, b)$ を満たす $x, y \in \mathbf{Z}$ が存在する。

系 4　$a, b \in \mathbf{Z}$ が互いに素であれば、$ax + by = 1$ を満たす $x, y \in \mathbf{Z}$ が存在する。

補題 5　$\mathbf{Z}$ の空でない部分集合 M が「差について閉じている」、つまり、条件

$$\forall\, a, b \in M, \quad a - b \in M$$

を満たしているとする。このとき、或る M の元 m が存在して M は $m\mathbf{Z}$ に一致する。

補題 5 の証明

Step 1 : $0 \in M$ であること。

$\because$　$M \neq \emptyset$ ゆえひとつ元 $a \in M$ を取れる。条件より $0 = a - a \in M$.

Step 2 : $a \in M$ ならば $a\mathbf{Z} \subseteq M$ であること。

$\because$　主張 1 と条件から、$-a = 0 - a \in M$. 以下、帰納的に $-2a = (-a) - a \in M$, $-3a = (-2a) - a \in M, \cdots$ が示され、$-ka \in M$ ($\forall\, k \in \mathbf{Z}$, $k \geqq 0$). 再び条件より $ka = 0 - (-ka) \in M$.

Step 3. $M = \{\, 0 \,\}$ のときは $m = 0$ として補題が成立するので問題なし。

以下、$M \neq \{\, 0 \,\}$ とする。Step 2 より M には正の元が存在するので M の正の元の中で最小のものを m とする。このとき $M = m\mathbf{Z}$ となることを示そう。

x を M の任意の元とし、x を m で割った商と余りを q, r と置く :

$$x = qm + r \quad (\, 0 \leqq r < m \,).$$

Step 2 より $qm \in M$ であるので、条件より

$$r = x - qm \in M.$$

もし $r > 0$ ならば m の最小性に反するので $r = 0$. すなわち M の任意の元は m の倍数になり、$M \subseteq m\mathbf{Z}$.

Step 2 より $m\mathbf{Z} \subseteq M$ でもあったので $M = m\mathbf{Z}$. (証明終)

定理 1 の証明

Step 1. $M = \{\, ax + by \,|\, x, y \in \mathbf{Z} \,\}$ と置くと、

$$(ax + by) - (ax' + by') = a(x - x') + b(y - y')$$

ゆえ M は補題 5 の条件を満たす。従って或る M の正の元 m が存在して $M = m\mathbf{Z}$. あとは $m = d$ であることを示せば良い。

Step 2. a, b は d の倍数ゆえ $ax + by$ の形の数も d の倍数。特に m も d の倍数ゆえ $m \geqq d$.

Step 3. $a = a \times 1 + b \times 0 \in M$, $b = a \times 0 + b \times 1 \in M$ ゆえ a, b は m の倍数。言い換えれば m は a, b の公約数になる。従って $d \geqq m$.

Step 2 と合わせて $d = m$. (証明終)

例 $a = 5$, $b = 13$ のとき M の元を書き出してゆくと、

$$\begin{aligned} 5 &= 5 \times 1 + 13 \times 0 \\ 13 &= 5 \times 0 + 13 \times 1 \\ 8 &= 5 \times (-1) + 13 \times 1 \\ 3 &= 5 \times (-2) + 13 \times 1 \\ -2 &= 5 \times (-3) + 13 \times 1 \\ 1 &= 5 \times (-5) + 13 \times 2 \end{aligned}$$

1 が $ax + by$ の形に表せられることがわかったので $\forall k \in \mathbf{Z}$ は

$$k = 5 \times (-5k) + 13 \times (2k)$$

と表すことができる。

2 不定方程式 $ax + by = c$ を解くアルゴリズム

定理 6 $a, b, c \in \mathbf{Z}$ を与えたとき、不定方程式 $ax + by = c$ が整数解 $x, y \in \mathbf{Z}$ を持つための必要十分条件は c が $d = \gcd(a, b)$ の倍数になることである。

定理 7 $a, b, c \in \mathbf{Z}$ $(ab \neq 0)$ が与えられ、c が $d = \gcd(a, b)$ の倍数であるとする。$x_0, y_0 \in \mathbf{Z}$ を不定方程式 $ax + by = d$ のひとつの整数解とすると、不定方程式 $ax + by = c$ の一般解 x, y は

$$x = (c/d)x_0 + (b/d)k, \quad y = (c/d)y_0 - (a/d)k \qquad (\, k \in \mathbf{Z} \,)$$

で与えられる。

アルゴリズム 8　不定方程式 $ax + by = c$ の一般解

Step 1. ユークリッドのアルゴリズムにより $d = \gcd(a, b)$ を求める。

Step 2. c が d で割り切れなければ「解なし」と出力して終了。

Step 3. $a/d \to A$, $b/d \to B$, $c/d \to C$.

Step 4. $ax_0 + by_0 = d$ を満たす x_0, y_0 を一組求める。
（ $Ax_0 + By_0 = 1$ を満たす x_0, y_0、と言っても同じ。）

Step 5. $x = Cx_0 + Bk$, $y = Cy_0 - Ak$ （$k \in \mathbf{Z}$）を一般解として出力。

例　$15x + 39 = 12$ の一般解

Step 1. $d = \gcd(15, 39) = 3$

Step 2. $c = 12$ は $d = 3$ で割り切れるので Step3 へ。

Step 3. $A = 5$, $B = 13$, $C = 4$.

Step 4. 先の例より $x_0 = -5$, $y_0 = 2$.

Step 5. $x = -20 + 13k$, $y = 8 - 5k$ （$k \in \mathbf{Z}$）が一般解。

3　ユークリッドのアルゴリズム拡張版

アルゴリズム 8 の Step 4 はユークリッドのアルゴリズムを拡張することにより高速に実現できる。

アルゴリズム 9　(ユークリッドのアルゴリズム拡張版)

入力：$a, b \in \mathbf{Z}$

出力：$d = \gcd(a, b)$ と、$d = ax + by$ を満たす $x, y \in \mathbf{Z}$

Step 1. 3 つの数列 $\{ r_n \}$, $\{ x_n \}$, $\{ y_n \}$ を用意する。

Step 2. 初期値は $\begin{cases} r_0 = a, & x_0 = 1, & y_0 = 0 \\ r_1 = b, & x_1 = 0, & y_1 = 1 \end{cases}$

Step 3. $n = 1$.

Step 4. $r_n = 0$ ならば Step 6 へ。

Step 5. $\begin{cases} n + 1 \to n \\ q = r_{n-2} \text{ を } r_{n-1} \text{ で割った商} \\ r_n = r_{n-2} - q \times r_{n-1} \\ x_n = x_{n-2} - q \times x_{n-1} \\ y_n = y_{n-2} - q \times y_{n-1} \end{cases}$

Step 6. $\begin{cases} r_{n-1} \geqq 0 \text{ ならば} & d = r_{n-1}, & x = x_{n-1}, & y = y_{n-1} \\ \text{そうでなければ} & d = -r_{n-1}, & x = -x_{n-1}, & y = -y_{n-1} \end{cases}$

アルゴリズム 9 の証明　終了時の n で $d = |\, r_{n-1} \,|$ となることは前回示した。

$$r_n = ax_n + by_n \qquad (n = 0, 1, \cdots)$$

が成り立つことは帰納法により示される。

アルゴリズム 10　(ユークリッドのアルゴリズム拡張版再帰 ver.)
入出力は上に同じ。

Step 1. $b \neq 0$ ならば Step 3 へ。

Step 2. $\begin{cases} a \geqq 0 \text{ ならば} & d = a, & x = 1, & y = 0 \\ \text{そうでなければ} & d = -a, & x = -1, & y = 0 \end{cases}$
を出力して終了。

Step 3. $\begin{cases} q = a \text{ を } b \text{ で割った商} \\ r = a - b \times q \end{cases}$
とし、再帰呼び出しで
$\begin{cases} d_0 = \gcd(b, r) \text{ と} \\ d_0 = bx_0 + ry_0 \text{を満たす } x_0,\, y_0 \end{cases}$
を求め $d = d_0,\, x = y_0,\, y = x_0 - q \times y_0$ を出力。

アルゴリズム 10 の証明　$b = 0$ のとき (Step 2) は $d = |\, a \,| = a \times (\pm 1) + b \times 0$ ゆえ OK. $b \neq 0$ のときは帰納法の仮定により

$$d = d_0 = bx_0 + ry_0 = bx_0 + (a - bq)y_0 = ay_0 + b(x_0 - qy_0).$$

定理 11　アルゴリズム 9、アルゴリズム 10 の計算量は $O(\log^3 a)$ である。

注 12　再帰呼び出しの深さに制限がある処理系ではアルゴリズム 10 はエラーを起こす恐れがある (ホームページの `L03.c` の `euclid3`)。アルゴリズム 9 も、そのまま組んだのでは配列が必要になりメモリを喰う (同 `euclid1`)。そこでアルゴリズム 9 を、数列を 3 項ずつ使い回しするように組めば使用するメモリも少なくて良い (同 `euclid2`)。

注 13　アルゴリズム 8 では、Step 1 でユークリッドのアルゴリズム拡張版を使うことにより Step 4 が省略できる。