第73回数学教育実践研究会　レポート発表

余りものには『福』がある

北海道室蘭東翔高等学校教諭　長尾良平

平成22年6月12日　北海道大学情報教育館

1　始めに

昨年度は, 3年生の看護・医療系の数学（数学課題探求）を担当しており, 一通り~~調教~~演習を終えた後, トピック的な題材をいくつか扱った. その中から, **「ISBNコード」**についての実践について紹介したい.

ISBNコードは書籍に対して一意に決められる番号であり, **合同式**が応用されている. 新課程では数学Aに初等整数論の内容が設定されることもあり, 自身の学習の意味も込めて授業を行った.

2　授業の展開

授業では, まず合同式の定義を行い, 具体例を挙げた.

合同式とは・・・

整数a, bについて, nで割った時の余りが等しい時, a, bは「**nを法として合同**」といい,

$$a \equiv b \pmod{n}$$

と表す. $a \equiv b \pmod{n}$は「**$a-b$がnで割り切れること**」と言い換えられる.

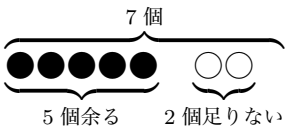
例1　定義の意味を具体例で確認.

(1) $17 \equiv 2 \pmod{5}$
- ☜ $17 \div 5 = 3 \cdots \mathbf{2}$
- ☜ $2 \div 5 = 0 \cdots \mathbf{2}$
- ☆ 17を5で割った余りは2

(2) $17 \not\equiv 4 \pmod{3}$
- ☜ $17 \div 3 = 5 \cdots \mathbf{2}$
- ☜ $4 \div 5 = 0 \cdots \mathbf{4}$
- ☆ 17を3で割った余りは4ではない

(3) $5 \equiv -2 \pmod{7}$
- ☜ $5 \div 7 = 0 \cdots \mathbf{5}$
- ☜ $-2 \div 5 = 0 \cdots \mathbf{5}$
- ☆ 「5余る」＝「2足りない」

(3)については,



という理解も必要だと考え, 図示することによって説明した.

これらの例を通して,（割り切れる部分はどうでもいいから）余りに注目していることを意識させた.

次に, 合同式の性質を紹介し, 余りを求める演習を通して, 合同式に慣れてもらった.

合同式の性質

$a \equiv b \pmod{n}$, $c \equiv d \pmod{n}$の時, 次が成り立つ.

- $a + c = b + d \pmod{n}$
- $a - c = b - d \pmod{n}$
- $ac = bd \pmod{n}$
- $a^k = b^k \pmod{n}$　kは自然数

例 2　$7 \equiv 2 \pmod 5$, $9 \equiv 4 \pmod 5$ より,

和　$16 \equiv 6 \pmod 5$　☜ 余り 1

積　$63 \equiv 8 \pmod 5$　☜ 余り 3

例 3　7^{50} を 4 で割った時の余りを求めよ.

$$7 \equiv -1 \pmod 4$$

の両辺を 50 乗して,

$$7^{50} \equiv (-1)^{50} \equiv 1 \pmod 4$$

よって, 7^{50} を 4 で割った時の余りは 1 である.

ISBN コードって知ってる？

ISBN ○-●●-△△△△△△-☆

- 「○」の部分で地域を表す.
- 「●●」で出版社を表す.
- 「△△△△△△」の部分は出版物固有の番号である.
- 「☆」の部分は**チェックディジット**.
  ☜ 入力時の誤りを確かめるためのもの.
  ☜ 0〜9, X（10 の代わりに）を当てる.

チェックディジットの定め方

ISBN コードで現れる各桁の数字を右から, $a_1, a_2, \ldots, a_{10}$ とする時,

$$\sum_{k=1}^{10} a_k \cdot k \equiv 0 \pmod{11}$$

が成り立つように, チェックディジット a_1 の値を定める.

「JR 全車両ハンドブック 2005」や「工場萌え」, 「Topology From The Differentiable Viewpoint」などの書籍を教室に持ち込み, ISBN コードについて説明した.

なお, 2007 年より規格が変わり, 現行の ISBN コードは 13 桁であり, チェックディジットの計算方法も変更になった.

例 4　「ISBN4-00-021326-1」が誤りを含むか否かをチェックせよ.

ISBN	4	0	0	0	2	1	3	2	6	1
重み	10	9	8	7	6	5	4	3	2	1
×	40	0	0	0	12	5	12	6	12	1

$$\begin{aligned} & 40 + 12 + 5 + 12 + 6 + 12 + 1 \\ \equiv\ & -4 + 1 + 5 + 1 - 5 + 1 + 1 \pmod{11} \\ \equiv\ & 0 \pmod{11} \end{aligned}$$

となり, 誤りを含んでいないことが分かる.

例 5　「ISBN4-00-005614-7」が誤りを含むか否かをチェックせよ.

ISBN	4	0	0	0	0	5	6	1	4	7
重み	10	9	8	7	6	5	4	3	2	1
×	40	0	0	0	0	25	24	3	8	7

$$\begin{aligned} & 40 + 25 + 24 + 3 + 8 + 7 \\ \equiv\ & -4 + 3 + 2 + 3 - 3 - 4 \pmod{11} \\ \equiv\ & -3 \pmod{11} \end{aligned}$$

となり, 誤りを含んでいることが分かる. なお, **誤りを検出できるが, 訂正はできない**（正しいコードは,「ISBN4-00-0056**41**-7」である）.

例 6　本に醤油をこぼしてしまって, ISBN コードの一部が見えなくなってしまった.

「ISBN4-00-02141 ●-6」であるとき, ●の部分の数字を答えよ.

●を a_2 で表すことにすると,

ISBN	4	0	0	0	2	1	4	1	a_2	6
重み	10	9	8	7	6	5	4	3	2	1
×	40	0	0	0	12	5	16	3	$2a_2$	6

$$\begin{aligned} & 40 + 12 + 5 + 16 + 3 + 2a_2 + 6 \\ \equiv\ & -4 + 1 + 5 + 5 + 3 + 2a_2 - 5 \pmod{11} \\ \equiv\ & 2a_2 + 5 \pmod{11} \end{aligned}$$

これより,

$$2a_2 + 5 \equiv 0 \pmod{11}$$

を満たす a_2 $(0 \leqq a_2 \leqq 9)$ を考えると, $a_2 = 3$ が見つかる（$2 \times 3 + 5 = 11$ より）.

3　授業を終えて

「大切なものは失って初めて気づく」のが世の常であるが, ISBN コードでは入力時の

- 任意の 1 箇所のミス
- 任意の 2 箇所の入れ替わり

を検出することができ（**訂正はできない**）,

- 任意の 1 箇所について**消失訂正**

まで行うことができる.

その理屈を初めて知った時,**「上手いこと考える人がいるもんだ！」**と感心した記憶がある. 自分が数学の教員を志したのは,**「自分が面白いと感じたものを伝えたい！」**という思いからであり, ISBN コードについてもどこかで扱いたいと考えていた.

授業時の生徒の反応は概ね良好で, 面白がって問題に取り組んでくれた.**「誰かに教えたい雑学になりました！」**という感想もあった.

ISBN コードは題材として,

- 簡単すぎず, 難しすぎない
- 意外性がある
- 身の回りにある

という点で, 優れていると考える. また, 生徒のレベルや扱う時間数, 時期によって

- 性質を確認し, 具体例で遊ぶ
- 理論的な考察を行う
- 他の例についても触れる

など, 様々な取扱いが可能である. 今回の実践では, 整数論入門ということで, 互除法・一次不定方程式（4 時間）, 合同式・ISBN コード（4 時間）という配当で行った.

他の例については, 最後に 2 つ挙げたので, 参考にしていただきたい. 何れも, コンピュータ関連で実際に使われているものである.

理論的な考察をするのであれば, ISBN の持つ性質を証明させてみるというのもよいだろう. その例を 1 つ挙げる.

定理　ISBN コードは, 任意の 2 箇所の数字が入れ替わってしまった入力誤りを検出できる.

すなわち,

$$a_k = b_k \quad (k \neq i, j)$$

$$a_i = b_j, a_j = b_i \quad (i \neq j)$$

のとき,

$$\sum_{k=1}^{10} a_k \cdot k \not\equiv \sum_{k=1}^{10} b_k \cdot k \pmod{11}$$

が成り立つ.

証明　$a_k = b_k$ $(k \neq i, j)$ であることより,

$$\begin{aligned} & \sum_{k=1}^{10} a_k \cdot k - \sum_{k=1}^{10} b_k \cdot k \\ = \ & (a_i \cdot i + a_j \cdot j) - (b_i \cdot i + b_j \cdot j) \\ = \ & (a_i \cdot i + a_j \cdot j) - (a_j \cdot i + a_i \cdot j) \\ = \ & a_i(i-j) - a_j(i-j) \\ = \ & (a_i - a_j)(i-j) \end{aligned}$$

ここで, $1 \leqq i, j \leqq 10$, $i \neq j$ より,

$$0 < |i-j| \leqq 9 \tag{1}$$

が成り立ち, 同様に, $0 \leqq a_i, a_j \leqq 10$, $a_i \neq a_j$ より,

$$0 < |a_i - a_j| \leqq 10 \tag{2}$$

が成り立つ. (1)(2) と, 11 が素数であることより, $(a_i - a_j)(i-j)$ は 11 の倍数とはなり得ない.

よって,

$$\sum_{k=1}^{10} a_k \cdot k - \sum_{k=1}^{10} b_k \cdot k \not\equiv 0 \pmod{11}$$

すなわち

$$\sum_{k=1}^{10} a_k \cdot k \not\equiv \sum_{k=1}^{10} b_k \cdot k \pmod{11}$$

であることが示された.

Q.E.D

4　終わりに

データに情報を付け加え, 安全性・正確性を向上させることを**冗長化**という. 実例を 2 つ挙げるが, ともに, mod2 で演算を行う. 排他的論理和（XOR：⊕）を考えると言ってもよい.

例 7（パリティ）　サーバ等で用いられる **RAID** では, パリティを作成し冗長化を行うものがある.

Data1
0
0
1
1

⊕

Data2
0
1
0
1

⇒

Parity
0
1
1
0

事故により, 1 台のデータが欠損してしまうと

Data1
☠
☠
☠
☠

Data2
0
1
0
1

Parity
0
1
1
0

となるが, 残りの 2 台で排他的論理和を考えると

Data1
0
0
1
1

⇐

Data2
0
1
0
1

⊕

Parity
0
1
1
0

となり, 欠損したデータの復元ができる.

例 8（ハミング符号）　誤り検出能力と併せて, **誤り訂正能力**も持っている.

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$H = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

をそれぞれ, 生成行列, パリティ検査行列という.

$\boldsymbol{u} = [1\ 0\ 0\ 1]$ を $\boldsymbol{v} = \boldsymbol{u}G$ によって符号化する.

$$\boldsymbol{v} = \boldsymbol{u}G = [1\ 0\ 0\ 1\ \mathit{1}\ \mathit{0}\ \mathit{0}]$$

となる. 右側の $[\mathit{1}\ \mathit{0}\ \mathit{0}]$ の部分が, 誤り訂正のために追加された情報である.

誤り検出・訂正は, $\boldsymbol{v}H^{\mathrm{T}}$ によって判断できる. $\boldsymbol{v}$ が誤りを含まなければ,

$$\boldsymbol{v}H^{\mathrm{T}} = \boldsymbol{O}$$

が成り立つ. 誤りを含んでいれば, $\boldsymbol{v}H^{\mathrm{T}} \neq \boldsymbol{O}$ となり, 誤りを検出し訂正することができる.

$\boldsymbol{v}$ の一部を改変した, $\boldsymbol{v}' = [1\ 0\ \mathbf{1}\ 1\ 1\ 0\ 0]$ で考えてみると,

$$\boldsymbol{v}'H^{\mathrm{T}} = [\mathbf{0}\ \mathbf{1}\ \mathbf{1}]$$

となり, H の 3 列目と一致している. このことは, $\boldsymbol{v}'$ の 3 番目が誤りであることを示している.

現行の ISBN コードを紹介して, 本稿を終える.

現行の ISBN コード

ISBN □□□ - ○ -●●-△△△△△△-☆

- 「□□□」は 978 または 979 のいずれかである.
- 残りの部分は, 旧規格と同じである.

ISBN コードで現れる各桁の数字を右から, $a_1, a_2, \ldots, a_{13}$ とする時,

$$\sum_{k=1}^{7} a_{2k-1} + \sum_{k=1}^{6} a_{2k} \cdot 3 \equiv 0 \pmod{10}$$

が成り立つように, チェックディジット a_1 の値を定める.

つまり, 右から奇数番目の数字の和に, 偶数番目の数字に 3 を乗じたものの和を加え, 10 の倍数になるように定める.

（注）ISBN の ○ -●●-△△△△△△の各部分の桁数は, 地域や出版社によって変化する（現規格・旧規格ともに）.

参考文献

[1] G.A. ジョーンズ, J.M. ジョーンズ
「情報理論と符号理論」シュプリンガー・ジャパン

[2] 「数学セミナー 2004 年 6 月号」日本評論社

[3] 日本図書コード管理センター
http://www.isbn-center.jp/