計算機数学I (2019)
第6回

照井 章（筑波大学 数理物質系 数学域）
Akira Terui (Institute of Mathematics, University of Tsukuba)

第5回のまとめ

- Horner法による1変数多項式の評価
- Horner法のアルゴリズムと計算量
- 非負整数の2進・10進変換
- 小数・分数の2進・10進変換

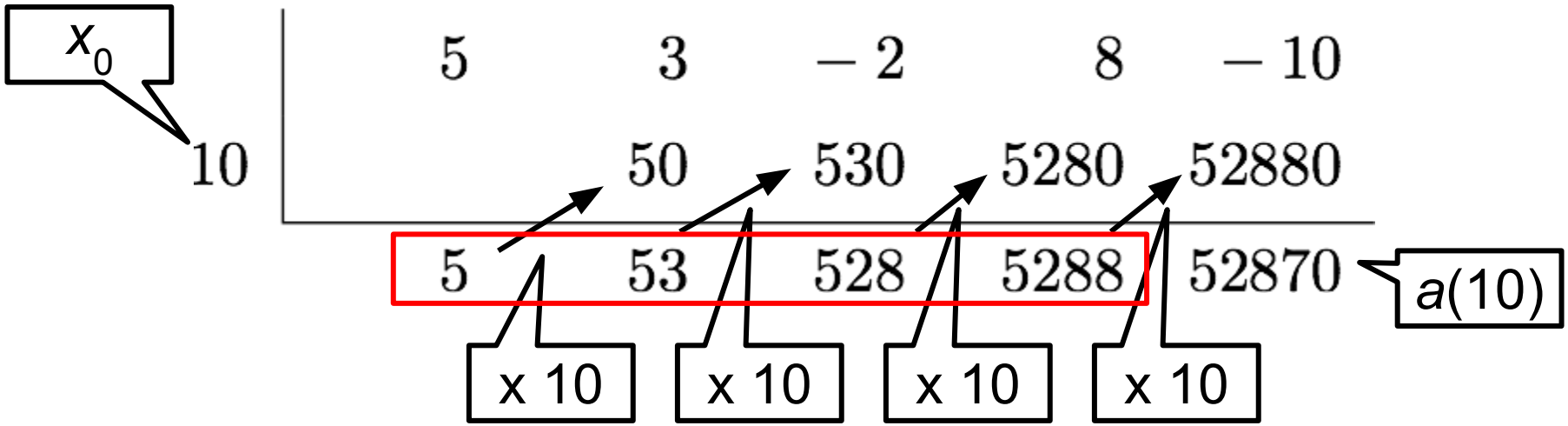
第6回の内容

- 多項式や多倍長整数の乗算
- 多項式の剰余つき除算

組立除法、Horner 法の補足

組立除法では実は商も同時に計算している

- 例題: $a(x) = 5x^4 + 3x^3 - 2x^2 + 8x - 10, \quad x_0 = 10$



$$a(10) = (((5 \cdot 10+3) \cdot 10-2) \cdot 10+8) \cdot 10-10 = 52870$$

- このとき、商は $5x^3 + 53x^2 + 528x + 5288$

2.3 多項式や多倍長整数の乗算 (p. 21)

2.3.1 1変数多項式の乗算

1変数多項式の乗算：例題

- $a(x) = a_3x^3 + a_2x^2 + a_1x + a_0 \in \mathbb{Z}[x], \quad a_3 \neq 0, \quad a_j \in \mathbb{Z}$
- $b(x) = b_1x + b_0 \in \mathbb{Z}[x], \quad b_1 \neq 0, \quad b_j \in \mathbb{Z}$

$$
\begin{array}{cccccc}
 & & a_3 & a_2 & a_1 & a_0 \\
\times) & & & & b_1 & b_0 \\
\hline
 & & a_3b_0 & a_2b_0 & a_1b_0 & a_0b_0 \\
 & a_3b_1 & a_2b_1 & a_1b_1 & a_0b_1 & \\
\hline
 & c_4 & c_3 & c_2 & c_1 & c_0
\end{array}
$$

1変数多項式の乗算:アルゴリズム

Algorithm 1変数多項式の乗算 $(a(x), b(x))$

入力: $a(x) = (a_0, a_1, \ldots, a_k) \in \mathrm{Z}[x],\quad a_k \neq 0,\quad a_j \in \mathrm{Z}$
$b(x) = (b_0, b_1, \ldots, b_m) \in \mathrm{Z}[x],\quad b_m \neq 0,\quad b_j \in \mathrm{Z}$

出力: $c(x) = (c_0, c_1, \ldots, c_{k+m}) \in \mathrm{Z}[x]$ s.t. $\mathrm{c}(x) = \mathrm{a}(x) \cdot \mathrm{b}(x)$

1. for $i \in [0..k+m]$ do $c_i \leftarrow 0$;
2. for $i \in [0..k]$ do
   a. for $j \in [0..m]$ do $c_{i+j} \leftarrow c_{i+j} + a_i\ b_j$;
3. return $(c_0, c_1, \ldots, c_{k+m})$;

1変数多項式の乗算：計算量の見積もり（整数の加減乗除を単位とする場合）

ループ $k+1$ 回

ループ $m+1$ 回

代入は無視

加算1回, 乗算1回

合計: $2(k+1)(m+1) = O(km)$

入力: $a(x) = (a_0, a_1, \ldots, \quad) \in \mathbb{Z}[x]$ $a_k \neq 0$ $a_k \in \mathbb{Z}$

$b(x) = (b_0, b_1, \ldots \quad b_m) \in \mathbb{Z}[x$ $\in \mathbb{Z}$

出力: $c(x) = (c_0, c_1, \ldots \quad c_{k+m}) \in \mathbb{Z}[x]$ s.t. $c(x) = a(x) \cdot b(x)$

1. for $i \in [0..k+m]$ do $c_i \leftarrow 0$;
2. for $i \in [0..k]$ do
   a. for $j \in [0..m]$ do $c_{i+j} \leftarrow c_{i+j} + a_i b_j$;
3. return (

1変数多項式の乗算：計算量の見積もり

- 演算回数は $a(x)$ と $b(x)$ の次数の積に比例
- しかし、ワード単位の計算量の見積もりは、次に述べる多倍長整数の乗算の見積もりが必要

2.3.2 多倍長整数と単精度整数の乗算

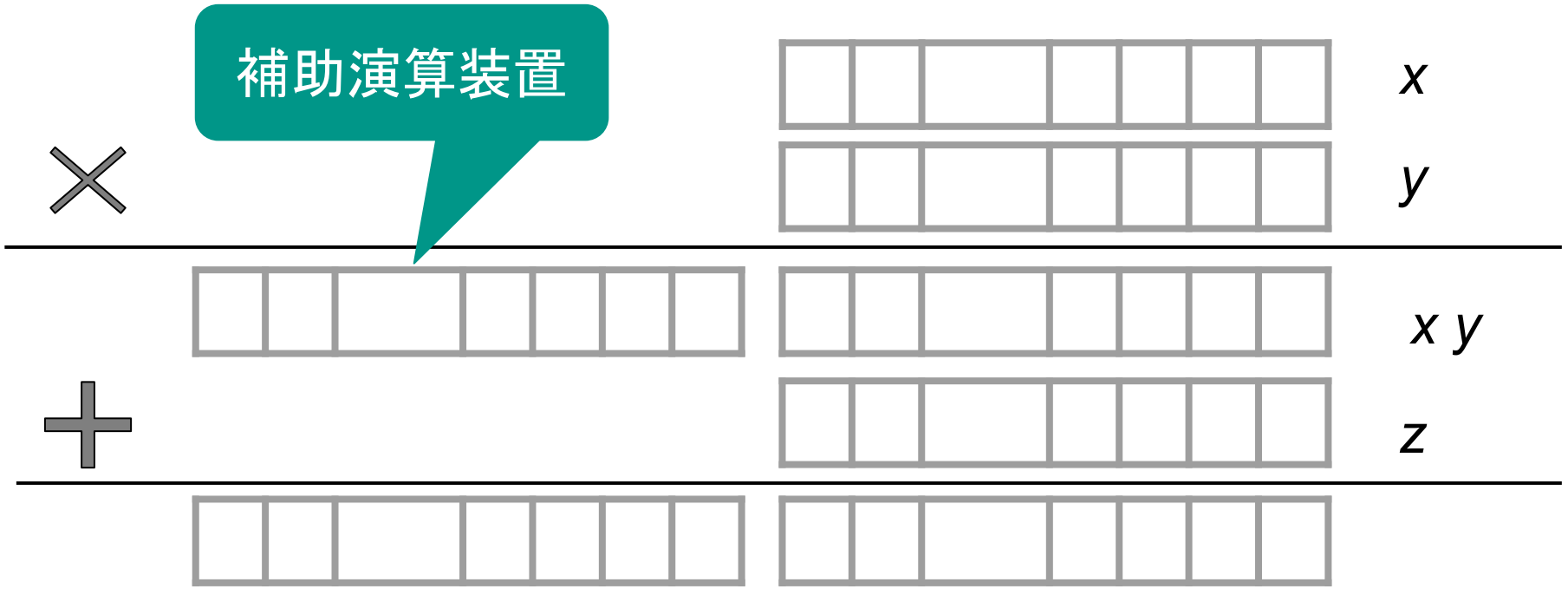
単精度整数どうしの乗算

- $x, y \in \mathbb{Z}$

x	×	y	=	xy
(n桁)		(n桁)		($2n$桁)
1ワード		1ワード		2ワード

分のメモリが必要

単精度整数どうしの乗算

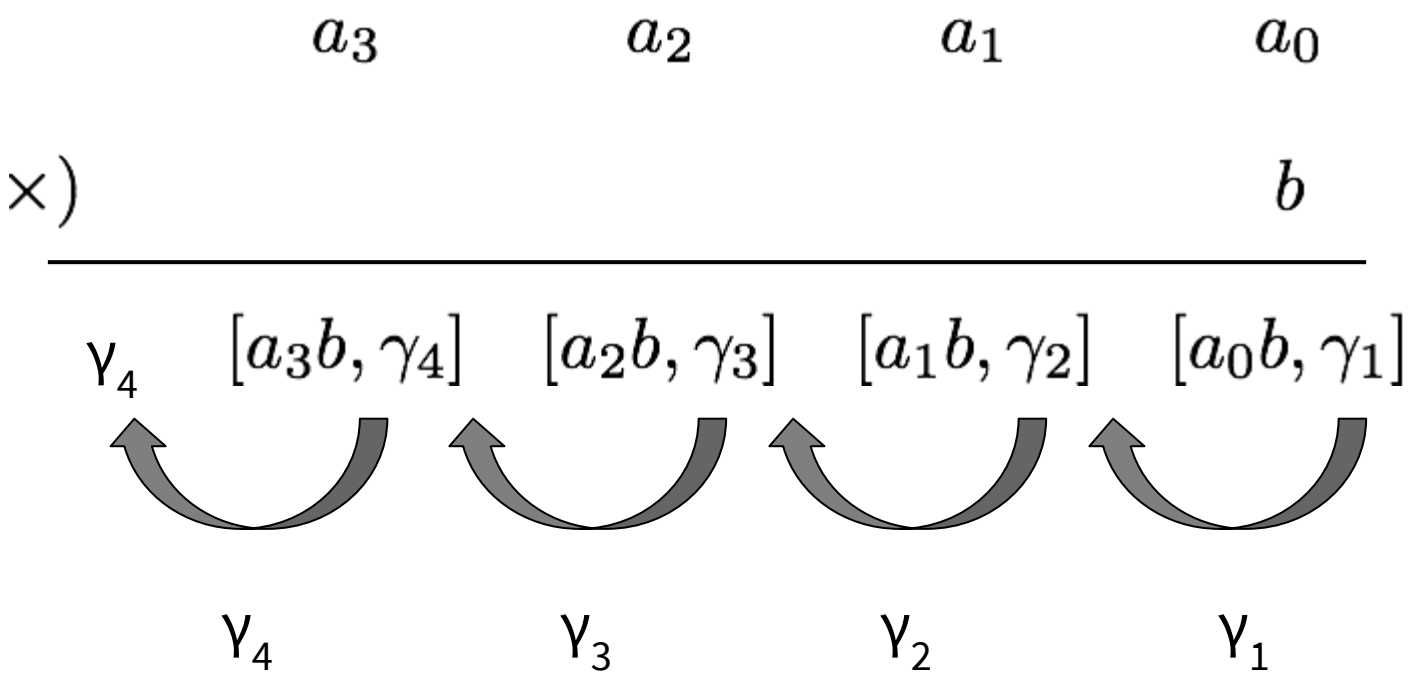


このとき $x\,y + z = \gamma \cdot 2^n + w$

これを $[w, \gamma] \leftarrow x\,y + z$ で表す

多倍長整数と単精度整数の乗算:例題

- $a = [a_0, a_1, a_2, a_3]$, b



多倍長整数と単精度整数の乗算:アルゴリズム

Algorithm 多倍長整数と単精度整数の乗算 (a, b)

入力: $a = [a_0, a_1, \ldots, a_{k-1}]$, $a_{k-1} \neq 0, b \in [0..2^n\text{-}1]$

出力: $c = [c_0, c_1, \ldots, c_k]$ s.t. $c = a \cdot b$

1. $\gamma_0 \leftarrow 0$;
2. for $i \in [0..k\text{-}1]$ do $[c_i, \gamma_{i+1}] \leftarrow a_i b + \gamma_i$;
3. $c_k \leftarrow \gamma_k$;
4. return $[c_0, c_1, \ldots, c_k]$

多倍長整数と単精度整数の乗算:計算量

Algorithm 多倍長整数と単精度整数の乗算 (a, b)

入力: $a = [a_0, a_1, \ldots, a_{k-1}]$, $a_{k-1} \neq 0, b \in [0..2^n\text{-}1]$

出力: $c = [c_0, c_1, \ldots, c_k]$ s.t. $c = a \cdot b$

ループ k 回

乗算1回, 加算1回

1. $\gamma_0 \leftarrow 0$;
2. for $i \in [0..k\text{-}1]$ do $[{}_ic, \gamma_{i+1}] \leftarrow a_i b + \gamma_i$;
3. $c_k \leftarrow \gamma_k$;
4. return $[c_0, c_1, \ldots, c_k]$

合計: $O(k) = O(\mathrm{len}(a))$

2.3.3 多倍長整数どうしの乗算

多倍長整数どうしの乗算：例題

- $a = [a_0, a_1, a_2, a_3, a_4]$, $b = [b_0, b_1]$

$$
\begin{array}{rccccccc}
 & & a_4 & a_3 & a_2 & a_1 & a_0 & \\
\times) & & & & & b_1 & b_0 & \\
\hline
 & & c'_5 & a_4b_0 & a_3b_0 & a_2b_0 & a_1b_0 & a_0b_0 \\
 & c'_6 & a_4b_1 & a_3b_1 & a_2b_1 & a_1b_1 & a_0b_1 & \\
\hline
 & c_6 & c_5 & c_4 & c_3 & c_2 & c_1 & c_0
\end{array}
$$

多倍長×単精度

多倍長×単精度

多倍長×多倍長

多倍長整数どうしの乗算:アルゴリズム

Algorithm 多倍長整数どうしの乗算 (a, b)

入力: $a = [a_0, a_1, \ldots, a_{k-1}], a_{k-1} \neq 0, b \in [b_0, b_1, \ldots, b_{m-1}], b_{m-1} \neq 0$

出力: $c = [c_0, c_1, \ldots, c_{k+m-1}]$ s.t. $c = a \cdot b$

1. $[c_0, c_1, \ldots, c_k] \leftarrow$ 多倍長・単精度乗算(a, b_0);
2. for $i \in [1..m\text{-}1]$ do
   $[c_i, \ldots, c_{k+i}] \leftarrow [c_i, \ldots, c_{k+i-1}]$
   + 多倍長・単精度乗算(a, b_i);
3. return $[c_0, c_1, \ldots, c_{k+m-1}]$

多倍長整数どうしの乗算:計算量

Algorithm 多倍長整数どうしの乗算 (a, b)

入力: $a = [a_0, a_1, \ldots, a_{k-1}], a_{k-1} \neq 0, b \in [b$ $\neq 0$

出力: $c =$ s.t. $c = a \cdot b$

乗算 O(len(a))

ループ m-1 回

1. $[c_0, c_1, \ldots, c_k] \leftarrow$ 多倍長・単精度乗算(a, b_0);
2. for $i \in [1..m\text{-}1]$ do
   $[c_i, \ldots, c_{k+i}] \leftarrow [c_i, \ldots, c_{k+i-1}]$
   + 多倍長・単精度乗算(a, b_i);
3. return $[c_0, c_1, \ldots, c_{k+m-1}]$

加算 O(len(a))

乗算 O(len(a))

合計: O(m len(a)) = O(len(a) len(b))

2.4 剰余つき除算

整数の剰余つき除算

- 定理（整数の剰余つき除算の性質）
  - $a, b \in \mathbb{Z}, b > 0$ のとき

    $\exists q, r \in \mathbb{Z}$ s.t. $a = bq + r$
  - a の b による除算: $a \div b$

整数の剰余つき除算

- 定義（整数の剰余つき除算）
  - a, b, q, r : 前の定理の通り
  - a: 被除数 (dividend)
  - b: 除数 (divisor)
  - q: 商 (quotient) = quo(a, b)
  - r: 剰余 (remainder) = a mod b
  - 商と剰余を同時に求める計算:
    $(q, r) \leftarrow a \div b$

2.4.1 多項式の剰余つき除算

多項式の剰余つき除算

- $a(x), b(x) \in \mathbb{Z}[x]$, $b(x)$: モニック
- $\deg(a) = k \geqq m = \deg(b)$

に対し、多項式の剰余つき除算を行う

多項式の剰余つき除算:アルゴリズム

Algorithm 多項式の剰余つき除算 (a, b)

入力: $a(x) = (a_0, a_1, \ldots, a_k) \in \mathrm{Z}[x], \quad a_k \neq 0, \quad a_j \in \mathrm{Z}$

$b(x) = (b_0, b_1, \ldots, b_{m-1}, 1) \in \mathrm{Z}[x], \quad b_j \in \mathrm{Z},$

$m \leqq k$

出力: $q(x) = (q_0, q_1, \ldots, q_{k-m}) \in \mathrm{Z}[x],$

$r(x) = (r_0, r_1, \ldots, r_{m-1}) \in \mathrm{Z}[x]$

s.t. $a(x) = q(x) \cdot b(x) + r(x)$

多項式の剰余つき除算:アルゴリズム

Algorithm 多項式の剰余つき除算 (a, b)

1. for $i \in$ [k..0] do $r_i \leftarrow a_i$;
2. for $i \in$ [k-m..0] do
   a. $q_i \leftarrow r_{i+m}$;$r_{i+m} \leftarrow 0$;
   b. for $j \in$ [m-1..0] do
      $r_{i+j} \leftarrow r_{i+j} - q_i b_j$;
3. return $(q_0, q_1, \ldots, q_{k-m}), (r_0, r_1, \ldots, r_{m-1})$

2.4.2 多倍長整数の剰余つき除算（省略）

第6回のまとめ

- 1変数多項式の乗算
- 多倍長整数と単精度整数の乗算
- 多倍長整数どうしの乗算
- 整数の剰余つき除算
- 多項式の剰余つき除算

第7回の内容

- 第4章: ユークリッド互除法

  (The Euclidean Algorithm)