数理解析研究所講究録
第1814巻 2012年 61-69

最短加減算連鎖生成のアルゴリズムについて

立教大学・理学研究科　内田　貴博 (Takahiro UCHIDA)
立教大学・理学研究科　小副川　健 (Takeshi OSOEKAWA)
情報通信研究機構　篠原　直行 (Naoyuki SHINOHARA)

概要

正の整数 m の加減算連鎖とは, $a_i = a_j \pm a_k$ $(\exists j, \exists k < i)$ を満たす数列 $1 = a_0, a_1, \ldots, a_r = m$ である. 本稿の目的は, 与えられた m に対して長さが最小の加減算連鎖を生成する効率の良いアルゴリズムを設計することである. そのために加減算連鎖に正規形の概念を導入し, 無駄な計算を省くことで計算効率の向上を図る.

1 はじめに

1894年に H.Dellac[1] が「x の $m \in \mathbb{N}$ 乗を計算する際に必要な最小の乗算回数は何回であるか？」という問題を提唱した. この問題は Dellac に提唱されて以来, 100年以上にわたり研究され, 様々な結果が報告されている [2]. x のベキ乗 x^m, x^n の乗算は $x^m \cdot x^n = x^{m+n}$ と指数部分が加法になることから, Dellac の問題を抽象定式化した加算連鎖の概念が生まれた. この理論は, RSA 暗号などの有限体上の乗算が必要な場合において計算効率を上げるために導入される. 具体的には $a \in \mathbb{Z}_p^*$ の $m \in \mathbb{N}$ 乗を計算する際に必要な最小の乗算回数を解析することは有益である.

一方, 加減算連鎖は除算も許した場合に加算連鎖を拡張した概念である. 有限体上の除算にも加減算連鎖の概念を適用することは可能であるが, 乗算に比べて除算は計算コストが高いため, a^m の総計算量の低減を考える場合それを適用することは有効とは言えない. しかし楕円曲線暗号で必要となる楕円曲線上の加法演算では, 加算と減算の計算コストが等しいため, 計算コストの観点から加減算連鎖を適用することは有効である. そこで与えられた $m \in \mathbb{Z}_{\geq 0}$ に対する加減算連鎖での最短経路を求めたい. しかし現在のところ最短経路を計算する方法は全数検索しか知られていない. さらに全数探索では計算量が指数時間である. そのため加算連鎖の理論で定義されていた下界数列を加減算連鎖に拡張し, 新しく正規形の概念を導入することで計算コスト削減を行う.

この論文は以下のように構成されている. 2節では加算連鎖の理論を, 3節では加減算連鎖の理論について述べる. ここでの理論は次節以降の理論の重要な基礎となる. 4節では正規形の概念について述べる. 正規形の導入により, 探索木からの枝刈りを行う. 5節では下界数列の概念について述べる. 加算連鎖の理論で定義されていた下界数列を加減算連鎖に拡張し, 探索木から枝刈りを行う.

2 準備

この節では加算連鎖の概念について説明する. 加減算連鎖は加算連鎖を拡張したものであるため, 加算連鎖の理論の多くが加減算連鎖に拡張できる. 加算連鎖については Knuth[2] の 4.6 節を参照されたい.

定義 2.1 (加算連鎖)
重複のない正の整数の数列 $1 = a_0, a_1, \ldots, a_r = m$ に対して, 各要素 a_i が

$$a_i = a_j + a_k \quad (0 \leq \exists j, \exists k < i)$$

を満たすとき, 数列 $\langle a_i \rangle_{i=0}^{r}$ を **m の加算連鎖**という. また各 i を**ステップ**, r を加算連鎖の**長さ**という. □

一般性を失わず, 加算連鎖は昇順に並んでいると仮定できる. 今後ことわらない限り, 昇順に並んでいるとする. また長さ r は加算・2 倍算の総和である. 加算連鎖の概念を導入することで Dellac の問題は以下で定義される $l(m)$ を求める問題に帰着することができる.

定義 2.2 (最短の加算連鎖)
すべての m の加算連鎖の中で, 長さ r が最小となる加算連鎖を **m の最短の加算連鎖** といい, このときの長さを $l(m)$ と表す. □

この $l(m)$ を求める問題は複雑であり公式などの存在が知られていない. 現在のところ一般の正の整数 m に対して $l(m)$ を求める方法は全数探索する以外ない. 次に探索木により全数探索する方法を紹介する. 詳しくは Thurber[6] を参照されたい.

■探索木
与えられた正の整数 m に対して, $l(m)$ を求めるため, 探索木にて加算連鎖を表現する. 探索木は図 1 のように**節**と節の間を結ぶ一つの**枝**で構成されるグラフの一種であり, 一番上の節を**根**, 互いに異なる節の並びを**道**, 根からある節までの道に含まれる枝の数を**深さ**という.

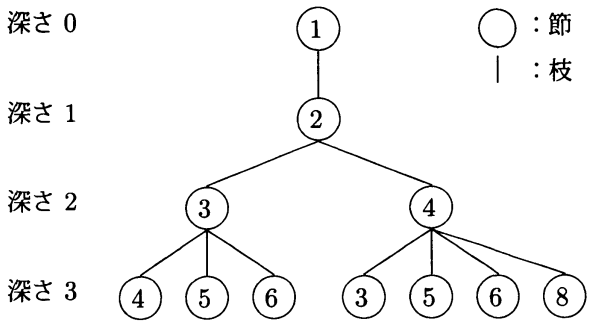

図 1 $l(m)$ を求めるための探索木

根とそれ以外の節にはただ一つの道があること注意する. ここでは各節に整数値を一つ格納しているとし, 根は 1 を格納しているものとする. 加算連鎖を探索木における道で表現する. 例えば, 根から深さ 2 の節までの道は加算連鎖 $\langle 1,2,3 \rangle$ と $\langle 1,2,4 \rangle$ であり, 長さが 2 である加算連鎖 $\langle a_i \rangle_{i=0}^{2}$ はこの 2 通りしかない. 同様に, 長さが 3 である加算連鎖 $\langle a_i \rangle_{i=0}^{3}$ は図 1 の探索木で表現されている 7 通りしかない. 各節に対して, 枝で結ばれている下方の節のことを**子節**という. ある節において加算連鎖 $\langle a_i \rangle_{i=0}^{r}$ が表現されていた場合, その子節として $a_{r+1} \in \{a_i \mid a_i = a_j + a_k \ (0 \leq j,k \leq r)\}$ かつ $a_{r+1} \neq a_0, a_1, \ldots, a_r$ を満たす, すべての a_{r+1} を枝で結んでいく. これにより加算連鎖 $\langle a_i \rangle_{i=0}^{r+1}$ を表現する. 記号として加算連鎖 $\langle a_i \rangle_{i=0}^{r}$ に対して

$$\mathrm{br}(\langle a_i \rangle_{i=0}^{r}) := \text{"探索木における加算連鎖 } \langle a_i \rangle_{i=0}^{r} \text{を表現している節までの道"}$$

とする.

■$l(m)$ を求めるためのアルゴリズム

探索木において次の手順により $l(m)$ を求めることができる. 詳しくは Thurber([6]) を参照.

Algorithm 1 $l(m)$ を求めるアルゴリズム

Require: 正の整数 m.

Ensure: $l(m)$.

1: $lb \leq l(m)$ を満たす lb を求める.
2: **loop**
3: 深さ lb まで探索木を展開する.
4: 深さ lb に m を格納している節があるか調べる.
あるならば, $l(m) = lb$ を出力する.
なければ, $lb \leftarrow lb + 1$ とする.
5: **end loop**

注意 2.3

上記の方法を用いて, $l(m)$ を計算機上で求めることができるが, 計算量が指数時間 $O((e^{c \log m \log\log m}))$ になる. なぜならば深さ k の節において子節の数は最低 $k+1$ 個あるため, 深さ k のとき探索木全体の節の個数は $k!$ 個となり, 探索すべきノードの数は $\log_2 m$ に対して指数関数的に増加するためである.

3 加減算連鎖

この節では加減算連鎖を定義し, 最短の加減算連鎖を求める方法について議論する. まず加減算連鎖を加算連鎖を元に拡張する.

定義 3.1 (加減算連鎖)

重複がない正の整数の数列 $1 = a_0, a_1, \ldots, a_r = m$ に対して, 各要素 a_i が

$$a_i = a_j \circ a_k \quad (0 \leq \exists j, \exists k < i)$$

を満たすとき, 数列 $\langle a_i \rangle_{i=0}^{r}$ を **m の加減算連鎖** という. また各 i を**ステップ**, r を加減算連鎖の**長さ**という. ただし演算 $\circ$ は $a, b \in \mathbb{Z}_{\geq 0}$ に対して

$$a \circ b = \begin{cases} a + b, \\ | a - b | \end{cases}$$

を意味する. □

注意 3.2

加算連鎖とは違い, 加減算連鎖は昇順に並んでいると仮定できないことに注意する. 例えば加減算連鎖 $\langle 1, 2, 4, 8, 7 \rangle$ を昇順に並び替えた $\langle 1, 2, 4, 7, 8 \rangle$ は加減算連鎖ではないからである. □

定義 3.3 (最短の加減算連鎖)

すべての m の加減算連鎖 $\langle a_i \rangle_{i=0}^{r}$ の中で, 長さ r が最小となる加減算連鎖を **m の最短の加減算連鎖** といい, このときの長さを $\tilde{l}(m)$ と表す. □

正の整数 m に対する最短の加算連鎖を表す $l(m)$ とは異なるので注意する．この $\tilde{l}(m)$ を求める問題は，「x の $m \in \mathbb{N}$ 乗を計算する際に必要な最小の乗除算回数は何回であるか？」と言い換えることができる．

■ **$\tilde{l}(m)$ を求めるためのアルゴリズム**

任意の正の整数 m に対して，$\tilde{l}(m)$ を計算するアルゴリズムは全数探索しか知られていない．そのため $l(m)$ を求める際と同様に探索木を使用し全数探索をする．したがって加算連鎖は加減算連鎖に含まれるため，最短の加算連鎖を探索する際も同様に計算量が指数時間である．そこで加算連鎖の理論で定義されていた下界数列を加減算連鎖に拡張し，新しく正規形の概念を提案し枝刈りを行う．

4 加減算連鎖の正規形

3 節で述べたように，加減算連鎖は加算連鎖と違い昇順に並んでいると仮定することができない．よって全数探索を効率的に行うため，並び替えによって他の加減算連鎖と一致するものを取り除く工夫が必要である．そこで「正規形」の概念を導入し，加減算連鎖の代表元を定義することで探索する節の数を減らす．「正規形」は同値・順序によって一意的に定められる．

定義 4.1 (同値, 同値類)

$\langle a_i \rangle_{i=0}^{r}, \langle b_i \rangle_{i=0}^{r}$ を長さ r の加減算連鎖とする．2 つの加減算連鎖が適当な並び替えによって一致する，すなわち

$$a_i = b_j \quad (\forall a_i \in \langle a_i \rangle_{i=0}^{r}, \exists b_j \in \langle b_i \rangle_{i=0}^{r})$$

を満たすとき $\langle a_i \rangle_{i=0}^{r}, \langle b_i \rangle_{i=0}^{r}$ は **同値** であるといい，$\langle a_i \rangle_{i=0}^{r} \equiv \langle b_i \rangle_{i=0}^{r}$ で表す．
また同値である加減算連鎖の集合を**同値類** といい

$$[\langle a_i \rangle_{i=0}^{r}] := \{\langle b_i \rangle_{i=0}^{r} \mid \langle b_i \rangle_{i=0}^{r} \equiv \langle a_i \rangle_{i=0}^{r}\}$$

で表す．□

例として $\langle 1,2,3,4,5 \rangle \equiv \langle 1,2,4,3,5 \rangle \equiv \langle 1,2,3,5,4 \rangle$ である．この同値な加減算連鎖の集合に対して順序を導入する．

定義 4.2 (辞書式順序)

同値な加減算連鎖 $\langle a_i \rangle_{i=0}^{r}, \langle b_i \rangle_{i=0}^{r}$ に対して，$a_j < b_j$ $(j = \min\{i \mid a_i \neq b_i\})$ を満たすとき

$$\langle a_i \rangle_{i=0}^{r} \prec_{lex} \langle b_i \rangle_{i=0}^{r}$$

と表し，$\prec_{lex}$ を **辞書式順序 (lex 順序)** という．□

例として $\langle 1,2,3,4 \rangle \prec_{lex} \langle 1,2,4,3 \rangle$ である．

定理 4.3

任意の加減算連鎖 $\langle a_i \rangle_{i=0}^{r}$ に対して，対 $([\langle a_i \rangle_{i=0}^{r}], \prec_{lex})$ は全順序集合である．□

[証明] 省略 □

以下では加減算連鎖とは限らない数列 $\langle a_i \rangle_{i=0}^{r}, \langle b_i \rangle_{i=0}^{r}$ に対して，$a_j < b_j (j = \min\{i \mid a_i \neq b_i\})$ を満たすとき

$$\langle a_i \rangle_{i=0}^{r} \prec \langle b_i \rangle_{i=0}^{r}$$

と表すことにする．このとき $\prec$ は全順序である．

命題 4.4
任意の加減算連鎖 $\langle a_i\rangle_{i=0}^r$ に対して，対 $([\langle a_i\rangle_{i=0}^r], \prec_{lex})$ は整列集合である．□

[証明] $\{a_i\}_{i=0}^r$ を要素を昇順に並べた数列 $\langle x_i\rangle_{i=0}^r$ としたとき，$\{a_i\}_{i=0}^r$ の要素を並び替えて構成できる任意の数列 $\langle y_i\rangle_{i=0}^r$ に対して

$$\langle x_i\rangle_{i=0}^r \prec \langle y_i\rangle_{i=0}^r$$

が成り立つことから，$\{a_i\}_{i=0}^r$ の要素を並び替えて構成できる数列すべての集合を A としたときに，$\langle x_i\rangle_{i=0}^r$ は A の最小元となる．このことから

$$\langle x_i\rangle_{i=0}^r = X_0 \prec X_1 \prec X_2 \prec \cdots \prec X_s \quad (X_1, X_2, \ldots, X_s \in A)$$

となっている．$l = \min\{i \mid X_i$は加減算連鎖 $\}$ としたとき，X_l が $[\langle a_i\rangle_{i=0}^r]$ の最小元になっていることから，最小元が存在することがわかる．□

定義 4.5 (正規形)
$\langle a_i\rangle_{i=0}^r$ を長さ r の 加減算連鎖とする．$\exists\langle b_i\rangle_{i=0}^r \in [\langle a_i\rangle_{i=0}^r]$ に対して

$$\langle b_i\rangle_{i=0}^r \prec_{lex} \langle c_i\rangle_{i=0}^r \quad (\forall\langle c_i\rangle_{i=0}^r \in [\langle a_i\rangle_{i=0}^r] \setminus \{\langle b_i\rangle_{i=0}^r\})$$

を満たすとき $\langle b_i\rangle_{i=0}^r$ を $\langle a_i\rangle_{i=0}^r$ の**辞書式順序 $\prec_{lex}$ における正規形**といい，$\mathrm{nf}_{lex}(\langle a_i\rangle_{i=0}^r)$ と表す．□

例として $\langle 1,2,3,4\rangle = \mathrm{nf}_{lex}(\langle 1,2,4,3\rangle)$ であり，$\langle 1,2,4,8,7\rangle = \mathrm{nf}_{lex}(\langle 1,2,4,8,7\rangle)$ である．$\langle 1,2,4,7,8\rangle \neq \mathrm{nf}_{lex}(\langle 1,2,4,8,7\rangle)$ であることに注意する．命題 4.4 より，以下の補題 4.6 が導ける．

補題 4.6
任意の加減算連鎖 $\langle a_i\rangle_{i=0}^r$ に対して，$\mathrm{nf}_{lex}(\langle a_i\rangle_{i=0}^r)$ は必ず存在しそれは一意的である．□

[証明] 命題 4.4 より最小元が存在するため，$\mathrm{nf}_{lex}(\langle a_i\rangle_{i=0}^r)$ は必ず存在しそれは一意的である．□

補題 4.6 より，$\tilde{l}(m)$ を探索木で求めるには，正規形のみを考えれば十分であることがわかる．探索木における各深さでの節の数を (i) 正規形のみの場合 (ii) 全通りの場合の 2 通りを表 1 にまとめた．また各深さにおいて正規形を導入することにより，枝刈りできる割合もまとめた．深さ 11 の場合 98.8% の節を枝刈りすること

表 1　正規形より枝刈りできる割合

深さ	1	2	3	4	5	6	7	8	9	10	11
正規形	2	4	10	36	186	1304	11714	129964	1736238	27404073	503235843
全通り	2	4	11	50	370	4062	61508	1231216	31646256	1020983919	40591003142
枝刈りの割合	0%	0%	9.1%	28.0%	49.7 %	67.9 %	81.0 %	89.4 %	94.5 %	97.3 %	98.8 %

ができる．正規形の概念の導入により，多くの枝刈りを行うことができることがわかる．深くなるにつれて刈れる割合が増えることが期待できる．

5 下界数列

探索する節の数を減らすため下界数列を導入する．下界数列は加算連鎖に定義されていたものであるが，加減算連鎖の理論に拡張できる．以下では記号として

$$\max_j(\langle a_i\rangle_{i=0}^{r}) := \text{"加減算連鎖 } \langle a_i\rangle_{i=0}^{r} \text{の要素の中で } j \text{ 番目に大きい値"}$$

とする．このとき定理 5.3, 定理]5.5, 定理 5.7 は，Thurber[6] で対応する定理の証明において a_{r-j} を $\max_j(\langle a_i\rangle_{i=0}^{r-1})$ で置き換えることで得られる．

定義 5.1 (1 階下界数列 Thurber[6] の拡張)
$\langle a_i\rangle_{i=0}^{lb}$ を長さ lb である m の加減算連鎖，$\langle b_i\rangle_{i=0}^{lb}$ を数列とする．
このとき部分加減算連鎖 $\langle a_0, a_1, \ldots, a_k\rangle$ $(k < lb)$ に対して

$$\max_1(\langle a_i\rangle_{i=0}^{k}) < b_k \quad (k < lb)$$

が成り立つならば，部分加減算連鎖 $\langle a_0, a_1, \ldots, a_k\rangle$ がどのような連鎖を構成しようとも $a_{lb} \neq m$ であるとき，数列 $\langle b_i\rangle_{i=0}^{lb}$ を **1 階下界数列** という．□

このとき $\mathrm{br}(\langle a_i\rangle_{i=0}^{k})$ は m に達しないので探索木から刈ることができる．

定義 5.2 (下界数列 (A) Thurber[6])
m を正の整数とする．与えられた $lb \in \mathbb{Z}_{\geq 0}$ に対して

$$b_i = \lceil n/2^{lb-i} \rceil \quad (0 \leq i \leq lb)$$

と定義した数列 $\langle b_i\rangle_{i=0}^{lb}$ を 長さ lb における $\boldsymbol{m}$ **の下界数列 (A)** という．□

例として，長さ 4 における 10 の下界数列 (A) は $\langle 1, 2, 3, 5, 10\rangle$ である．下界数列 (A) は 1 階下界数列である．

定理 5.3 (Thurber[6] の拡張)
数列 $\langle b_i\rangle_{i=0}^{lb}$ を 長さ lb における m の下界数列 (A) とする．このとき加減算連鎖 $\langle a_i\rangle_{i=0}^{lb}$ に対して

$$\max_1(\langle a_i\rangle_{i=0}^{k}) < b_k \quad (k < lb)$$

ならば 部分加算減連鎖 $\langle a_0, a_1, \ldots, a_k\rangle$ はどのような連鎖を構成しようとも $a_{lb} \neq m$ である．よって $\mathrm{br}(\langle a_i\rangle_{i=0}^{k})$ は探索木から刈ることができる．□

[証明] $a_i \leq 2\max_1(\{a_0, a_1, \ldots, a_{i-1}\})$ であることから，$a_k > 2^m \cdot \max_1(\langle a_i\rangle_{i=0}^{j})$ を満たすならば，どんな連鎖を構成しようとも、a_j から a_k へは $m+1$ 以上のステップ数が必要である．よって，ある特定の長さ lb で m に達する加減算連鎖 $\langle a_i\rangle_{i=0}^{lb}$ を探しているとき，$2^{lb-j}a_j < m$ ならば、a_j から $lb-j$ ステップで m に到達するのは不可能である．つまり $a_j < m/2^{lb-j}$ ならば探索木から $\mathrm{br}(\langle a_i\rangle_{i=0}^{j})$ を刈ることができる．□

例として，長さ 4 における 10 の下界数列 (A) は $\langle b_i\rangle_{i=0}^{4} = \langle 1, 2, 3, 5, 10\rangle$ であるが，部分加減算連鎖 $\langle a_i\rangle_{i=0}^{3} = \langle 1, 2, 3, 4\rangle$ において，$a_3 < b_3$ であることから，$\mathrm{br}(\langle a_i\rangle_{i=0}^{3})$ は探索木から刈ることができる．下界数列 (A) は，m が奇数であるとき改良することができる．

定義 5.4 (下界数列 (B) Thurber[6])
m を正の奇数とする. 与えられた $lb \in \mathbb{Z}_{\geq 0}$ に対して

$$b_i = \begin{cases} \lceil m/(3 \cdot 2^{lb-(i+2)} \rceil & (0 \leq i \leq lb-2) \\ \lceil m/2^{lb-i} \rceil & (lb-1 \leq i \leq lb) \end{cases}$$

と定義した数列 $\langle b_i \rangle_{i=0}^{lb}$ を 長さ lb における ***m* の下界数列 (B)** という. □

例として, 長さ 5 における 13 の下界数列 (B) は $\langle 1,2,3,5,7,13 \rangle$ である.

定理 5.5 (Thurber[6] の拡張)
数列 $\langle b_i \rangle_{i=0}^{lb}$ を 長さ lb における m の下界数列 (B) とする. このとき加減算連鎖 $\langle a_i \rangle_{i=0}^{lb}$ に対して

$$\max_1(\langle a_i \rangle_{i=0}^{k}) < b_k \quad (k < lb)$$

ならば 部分加算減連鎖 $\langle a_0, a_1, \ldots, a_k \rangle$ はどのような連鎖を構成しようとも $a_{lb} \neq m$ である. よって $\mathrm{br}(\langle a_i \rangle_{i=0}^{k})$ は探索木から刈ることができる. □

[証明] 加減算連鎖 $\langle a_i \rangle_{i=0}^{lb}$ が m の最短の加減算連鎖であるとき, 必ず最後のステップは $a_{lb} = a_{lb-1} + a_k \quad (0 \leq k \leq lb-1)$ であり, m が奇数であるので $a_{lb} \neq 2a_{lb-1}$ である. よって

$$m \leq \max_1(\langle a_i \rangle_{i=0}^{lb-1}) + \max_2(\langle a_i \rangle_{i=0}^{lb-1}) \leq 3 \cdot \max_2(\langle a_i \rangle_{i=0}^{lb-1})$$

であるので $\max_2(\langle a_i \rangle_{i=0}^{lb-1}) \geq \lceil m/3 \rceil$ でない限り, lb 回のステップで m までたどり着くことができない. さらに $\lceil m/(3 \cdot 2) \rceil \leq \max_3(\langle a_i \rangle_{i=0}^{lb})$ でない限り, $\max_2(\langle a_i \rangle_{i=0}^{lb}) \geq \lceil n/3 \rceil$ にはならない. これより

$$\max(\{a_i\}) < \lceil n/(3 \cdot 2^{lb-(i+2)}) \rceil \quad (i = 0, 1, \ldots, lb-2)$$

を満たすとき, 探索木から $\mathrm{br}(\langle a_i \rangle_{i=0}^{k})$ を刈ることができることがわかる. □
例として, 長さ 5 における 13 の下界数列 (B) は $\langle b_i \rangle_{i=0}^{5} = \langle 1,2,3,5,7,13 \rangle$ であるが, 部分加減算連鎖 $\langle a_i \rangle_{i=0}^{4} = \langle 1,2,3,4 \rangle$ において, $a_3 < b_3$ であることから, $\mathrm{br}(\langle a_i \rangle_{i=0}^{3})$ は探索木から刈ることができる. 下界数列 (B) を m が偶数でも使用できるように, 一般化する.

定義 5.6 (下界数列 (C) Thurber[6])
$m = 2^t n$ (n は奇数, $t \geq 0$) を正の整数とする. 与えられた $lb \in \mathbb{Z}_{\geq 0}$ に対して

$$b_i = \begin{cases} \lceil m/(3 \cdot 2^{lb-(i+2)}) \rceil & (0 \leq i \leq lb-t-2) \\ \lceil m/2^{lb-i} \rceil & (lb-t-1 \leq i \leq lb) \end{cases}$$

と定義した数列 $\langle b_i \rangle_{i=0}^{lb}$ を 長さ lb における ***m* の下界数列 (C)** という. □

例として, 長さ 6 における 26 の下界数列 (C) は $\langle 1,2,3,5,7,13,26 \rangle$ である.

定理 5.7 (Thurber[6] の拡張)
数列 $\langle b_i \rangle_{i=0}^{lb}$ を 長さ lb における m の下界数列 (C) とする. このとき加減算連鎖 $\langle a_i \rangle_{i=0}^{r}$ に対して

$$\max_1(\langle a_i \rangle_{i=0}^{k}) < b_k \quad (k < lb)$$

ならば 部分加減算連鎖 $\langle a_0, a_1, \ldots, a_k \rangle$ は長さ lb の m の加減算連鎖を導くことができない. よって $\mathrm{br}(\langle a_i \rangle_{i=0}^{k})$ は探索木から刈ることができる. □

[証明] 上界数列 (C) を 2つの部分に分けて考える。

(i) $b_i = \lceil m/2^{lb-i} \rceil \quad (lb - t - 1 \leq i \leq lb)$
定理 5.3 より、$a_i < \lceil m/2^{lb-1} \rceil$ であれば $\mathrm{br}(\langle a_i \rangle_{i=0}^{k})$ は刈ることができる。

(ii) $b_i = \lceil m/(3 \cdot 2^{lb-(i+2)}) \rceil \quad (0 \leq i \leq lb - t - 2)$
$0 \leq i \leq lb - t - 2$ に対して $\max_1(\{a_0, \ldots, a_i\}) < \lceil m/(3 \cdot 2^{lb-(i+2)}) \rceil$ であると仮定する。このとき $\max_1(\{a_0, \ldots, a_i\}) < m/(3 \cdot 2^{lb-(i+2)})$ である。$s > i+1$ を満たすステップ s が $a_0, \ldots, a_{s-1}$ の 2 倍でないとすると、$\max_2(\{\langle a_i \rangle_{i=0}^{s-1}\}) \leq \max_1(\{\langle a_i \rangle_{i=0}^{s-2}\})$ かつ $\max_1(\langle a_i \rangle_{i=0}^{s}) \leq 2^{s-i}\max_1(\{a_0, \ldots, a_i\})$ であることから

$$\begin{aligned} a_s &\leq \max_1(\langle a_i \rangle_{i=0}^{s-1}\}) + \max_2(\{\langle a_i \rangle_{i=0}^{s-1}\}) \\ &\leq \max_1(\langle a_i \rangle_{i=0}^{s-1}\}) + \max_1(\langle a_i \rangle_{i=0}^{s-2}\}) \\ &\leq 2^{s-1-i}\max_1(\{a_0, \ldots, a_i\}) + 2^{s-2-i}\max_1(\{a_0, \ldots, a_i\}) \\ &= 2^{s-2-i}3\max_1(\{a_0, \ldots, a_i\}) \end{aligned}$$

であるがこれは

$$m = a_{lb} \leq 2^{lb-s}a_s \leq 2^{lb-s}2^{(s-2)-i}(3\max_1(\{a_0, \ldots, a_i\})) < 2^{lb-s}2^{(s-2)-i}(3)m/(3 \cdot 2^{lb-(i+2)}) = m$$

より矛盾である。また $i+1$ 以降のステップの各要素がすべてある要素の 2 倍だと仮定すると

$$m = a_{lb} = 2^{lb-(i+1)}a_{i+1}$$

であるが $i \leq lb - t - 2$ より $lb - i - 1 \leq t + 1$ であるので m が t 回より多く 2 で割れることになり、矛盾である。よって $\max_1(\{a_0, \ldots, a_i\}) < \lceil m/(3 \cdot 2^{lb-(i+2)}) \rceil$ であるという仮定が間違っている。□

例として, 長さ 6 における 26 の下界数列 (C) は $\langle a_i \rangle_{i=0}^{6} = \langle 1, 2, 3, 5, 7, 13, 26 \rangle$ であるが, 部分加減算連鎖 $\langle a_i \rangle_{i=0}^{4} = \langle 1, 2, 3, 5, 6 \rangle$ において, $a_4 \leq b_4$ であることから, $\mathrm{br}(\langle a_i \rangle_{i=0}^{4})$ は探索木から刈ることができる. 表 2 は深さ k において $\tilde{l}(m) = k$ を満たす m に対して, 正規形による枝刈りを行った後, さらに下界数列 (C) を導入することで枝刈りできる節の数の割合の平均をまとめたものである. 深さ 11 の場合 99.7% の節を枝刈りすることができる. 下界数列の導入により, 多くの枝刈りを行うことができることがわかる. 深くなるにつれて刈れる割合が増えることが期待できる.

表 2　各深さにおいて枝刈りできる節の数の割合

深さ	1	2	3	4	5	6	7	8	9	10	11
平均	0 %	0 %	10. 0 %	25. 0 %	40. 1 %	60. 8 %	78. 7 %	90. 1 %	96. 1 %	98. 8 %	99. 7 %

6　結果

長さが 12 の加減算連鎖に対する探索木の節の数を正規形の概念により約 98.8% カット, 下界数列により約 99.7% カット することができた. この 2 つの戦略を組み合わせることで, 長さが 12 の加減算連鎖に対する探索木の節の数を約 99.64 % カットすることに成功した.

7 今後の課題

- コストを加味した加減算連鎖を考えることで, 楕円曲線上のスカラー倍算の効率の良い計算方法として知られる NAF 法の改良の余地を検討する.
- 各ステップにおいて, 計算に利用できる要素を制限した加減算連鎖を考えることで, 記憶領域を考慮した場合の NAF 法の改良の余地を検討する.

謝辞

本稿を作成するに当たり, 立教大学の横山和弘教授に多大なアドバイスと御奨励を頂きました. ありがとうございました.

参考文献

[1] H.Dellac,Question 49,L'Intermdiaire des Mathmaticiens V.1,pp.20,1894.

[2] D.E.Knuth,The Art of Computer Programming V. 2 Seminumerical Algorithms,Addison-Wesly,pp.449-450,1969.

[3] A.Schoenhage,A Lower Bound for the Length of Addition Chains,Theoretical Computer Science V.1,pp.1-12,1975.

[4] K.B.Stolarsky,A lower bound for the Scholz-Brauer problem,Canadian Journal of Mathematics V. 21,pp.675-683,1969.

[5] E.G.Thurber,Addition chains - an eratic sequence,Discrete Mathematics V.122,pp.287-305,1993.

[6] E.G.Thurber,Efficient Generation of Minimal Length Addition Chains,SIAM Journal of Computing V.28,pp.1247-1263,1999.