

オンライン計算に於ける領域計算量

電通大 町田 元
笠井琢美

1. はじめに

計算量の理論(計算の複雑さの理論)に於いて, 具体的に与えられた何らかの問題に対して, その計算量(時間, 領域など)のよい下界を求めることは重要であり, また興味深い問題でもある。近年多くの研究がなされているが, 計算モデルとして通常のオフライン・チューリング機械を考えると, この問題は極めて難しく, 対角線論法のほかには有効な手法はほとんど知られていない。

これに対して, 計算モデルを, 入力テープのヘッドの動きを1方向のみに制限した, オンライン・チューリング機械に限定して考えると, いくつかの問題に対して計算量の自明でない下界を求めることが可能となる。Hartmanis と Shank ([4], [5]) はオンライン・チューリング機械のもとで素数の認識問題を考察し, この問題は領域量 n を必要とすることを

示した。また, Lewis, Stearns と Hartmanis [6] はある特定の文脈自由言語に対してその言語の認識問題の領域計算量が n であることを示し, Gallaire [3] は別の文脈自由言語についてその認識問題が $n^2/\log n$ 時間を必要とすることを示した。さらに, Paterson, Fischer と Meyer [7] はオンラインのほかはいくつかの制約のついた計算モデルの上で 2 進数の乗算に必要な時間計算量を考察した。

本論文では, このような流れを引きつぎ, いくつかの具体的に与えられた集合(問題)および関数について, オンライン・チューリング機械のもとでの領域計算量の下界について考察する。計算モデル(チューリング機械)に対する制約はオンラインであるということだけで, そのほかには特に制限を加えない。取り上げるおもな問題および関数は次のようなものである。

(i) 3 のべき乗を表わす 2 進列全体の集合。

(ii) 整数(2 進数)の乗算。

(iii) 中置記法で表わされた算術式の Polish 記法への変換。

主要結果として, (i), (ii), (iii) のいずれもちょうど n のオーダーの領域を必要とすることを示す。また, (i) に対する結果から直ちに, 2 進数を 3 進数に変換する関数も n の領域を必要とすることが導かれる。一方, (ii) の証明に関連して,

非決定性オンライン・チューリング機械で $O(\log n)$ 領域で認識可能な集合のクラスは，決定性オンライン・チューリング機械で $O(\log n)$ 領域で認識可能な集合のクラスを真に含むことなども示す。

本稿では，ページ数の制限のため，論旨の筋道に直接関係しない補題の証明は省略した。省略した証明のなかには，主定理の証明の本質的な部分を含むものもある。

2. 準備

まず，オンライン・チューリング機械， $S(n)$ 領域計算可能， $S(n)$ 領域必要，など本稿に於ける考察の基礎となる概念を定義する。

定義 オンライン・チューリング機械とは，入力テープ，出力テープ各1本と有限個の(半無限)補助テープとから成るチューリング機械で，入力テープと出力テープに対する各ヘッドの動きが1方向のみしか許されていないものことである(補助テープに対するヘッドはどちらの方向に動いてもよい)。

今後，チューリング機械は，オンライン・チューリング機械しか考えないので，単にチューリング機械と書いてもオンライン・チューリング機械を表わすこととする。

定義. Σ_1 と Σ_2 を任意のアルファベットとする。関数 $f: \Sigma_1^* \rightarrow \Sigma_2^*$ および $S: \mathbb{N} \rightarrow \mathbb{N}$ に対して、 f が $S(n)$ 領域計算可能 であるとは、次の (1) ~ (3) を満たすチューリング機械 M が存在することである。

- (1) M の入力アルファベットは Σ_1 、出力アルファベットは Σ_2 である。
- (2) 任意の入力 $w \in \Sigma_1^*$ に対して、 M は出力テープ上に $f(w)$ を出力して、停止する。
- (3) 任意の入力 $w \in \Sigma_1^*$ に対して、 w の長さ $|w|$ を n とおくと、 M のどの補助テープについても計算の過程で高々 $S(n)$ 個のセルしか用いられない。

また、 Σ_1^* の部分集合について、 A が $S(n)$ 領域認識可能 であるとは、 A の特性関数 χ_A が $S(n)$ 領域計算可能であることをいう。ただし、このとき $\Sigma_2 = \{0, 1\}$ で、出力は入力列をすべて読み終わったときにのみ与えられるものとする。

定義. f および S を上と同様とする。

$$\inf_{n \rightarrow \infty} \frac{S'(n)}{S(n)} = 0$$

を満たすいかなる関数 $S': \mathbb{N} \rightarrow \mathbb{N}$ についても f が $S'(n)$ 領域計算可能でないとき、 f は $S(n)$ の領域を必要とする（または、 f の領域計算量は $\Omega(S(n))$ である）という。また、集

合 A についても, A の特性関数 C_A が $S(n)$ の領域を必要とするとき, A は $S(n)$ の領域を必要とする (または, A の領域計算量は $\Omega(S(n))$ である) という。

定義 Σ_1 と Σ_2 を任意のアルファベットとする。部分集合 $A \subseteq \Sigma_1^*$, $B \subseteq \Sigma_2^*$ および関数 $f: \Sigma_1^* \rightarrow \Sigma_2^*$ について, 任意の $w \in \Sigma_1^*$ に対して,

$$w \in A \iff f(w) \in B$$

が成り立つとき, A は f によって B に 還元可能 であるという。

次の2つの補題は定義から直ちに得られる。

補題2.1 Σ をアルファベットとし, $A \subseteq \Sigma^*$ とする。このとき, A が正規集合であることと, A が $O(1)$ 領域認識可能であることは同値である。

補題2.2 Σ をアルファベットとし, A, B を Σ^* の部分集合とする。 A が $S_1(n)$ 領域認識可能であり, B が $S_2(n)$ 領域認識可能であれば, $A \cup B$ および $A \cap B$ はいずれも $S_1(n) + S_2(n)$ 領域認識可能である。

記法 これ以後, $\Sigma = \{0, 1\}$ とおく。また, Σ^+ の元 $w = a_1 a_2 \dots a_n$ ($n > 0$) に対して, w が表わす2進数 (ただし, 左端を低位, 右端を高位のビットとする) を $[w]$ と記す。すなわち,

$$[w] = \sum_{i=1}^n a_i \cdot 2^{i-1}$$

である。

3. 下界を求めるための手法

次節以後、いくつかの問題、関数についてその領域計算量の下界が n であることを示すが、それらの証明で用いられる基本的な考え方は次の補題で表わされる。

補題 3. 1 Γ をアルファベットとし、 A を Γ^* の任意の部分集合とする。各 $n \in \mathbb{N}$ に対して $A^{(n)}$ を集合

$$\{u \in \Gamma^* \mid |u| = n, \text{ ある } v \in \Gamma^* \text{ が存在して } uv \in A\}$$

の 1 つの部分集合とする。このとき、次の 2 条件が満たされるならば、集合 A は n の領域を必要とする。

(1) 定数 c および n_0 が存在して、任意の $n > n_0$ に対して、

$$|A^{(n)}| \geq 2^{n-c}.$$

(2) $A^{(n)}$ の中の任意の相異なる 2 元 u_1, u_2 に対して、 v ($\in \Gamma^*$) が存在して、

$$u_1 v \in A \text{ かつ } u_2 v \notin A$$

$$\text{または } u_1 v \notin A \text{ かつ } u_2 v \in A$$

略証。入力を途中まで読んだ時の情報量を評価することによって得られる。 □

一方、すでに下界が知られている集合を媒介にして還元可能性を用いて別の集合の下界を求めることもできる。そのと

き用いられる補題は次のものである。

補題 3. 2 Σ_1, Σ_2 をアルファベットとし, $A \subseteq \Sigma_1^*, B \subseteq \Sigma_2^*$
 $f: \Sigma_1^* \rightarrow \Sigma_2^*$ とする。A は B に f によって還元可能であり, B
 は $S_1(n)$ 領域認識可能, f は $S_2(n)$ 領域計算可能とすると,
 A は $S_1(n) + S_2(n)$ 領域認識可能である。

4. 3 のべき乗を表わす 2 進列の集合の認識

3 のべき乗を表わす 2 進列全体から成る集合を T とおく。

すなわち,

$$T = \{ w \in \Sigma^* \cdot 1 \mid [w] = 3^k, k \in \mathbb{N} \}.$$

T が $O(n)$ 領域認識可能であることは容易にわかる。本節の
 目的は, 次の定理を得ることである。

定理 4. 1 集合 T は n の領域を必要とする。

補題 3. 1 を用いて定理 4. 1 を証明する。任意の $k \in \mathbb{N}$
 に対して, 3^k を表わす 2 進列 ($\in \Sigma^* \cdot 1$) を w_k とおく。す
 なわち,

$$[w_k] = 3^k.$$

従って,

$$T = \{ w_k \mid k \in \mathbb{N} \}$$

となる。さらに, $n \in \mathbb{N}$ に対して, T の長さ n 以上の元 w_k
 の先頭 n 文字からなる列を $w_k^{(n)}$ とし, それらの列全体から

なる集合を $T^{(n)}$ とおく。従って,

$$T^{(n)} = \{ W_k^{(n)} \mid k \in \mathbb{N}, |W_k^{(n)}| = n, (\exists v \in \Sigma^*) W_k^{(n)} \cdot v = W_{k+1} \}$$

$T^{(n)}$ の元が表わす整数を $(k$ の) 順に並べた数列

$$[W_{k_n}^{(n)}], [W_{k_{n+1}}^{(n)}], [W_{k_{n+2}}^{(n)}], \dots$$

を考えると, これは周期列である(ここで, k_n は $|W_k| \geq n$ を満たす最小の k である)。この数列の周期を $\pi(n)$ とおく。

すなわち, 任意の $n \in \mathbb{N}$, $k \geq k_n$ に対して,

$$[W_{k+\pi(n)}^{(n)}] = [W_k^{(n)}]$$

であるが, この $\pi(n)$ について次の補題が成り立つ。

補題 4. 2 任意の $n \geq 3$ に対して, $\pi(n) = 2^{n-2}$.

証明. 略.

系. $n \geq 3$ のとき, $|T^{(n)}| = 2^{n-2}$.

定理 4. 1 の証明. T , $T^{(n)}$ を各々補題 3. 1 に於ける A ,

$A^{(n)}$ とする。上の系より $T^{(n)}$ が (1) を満たすことが示される。

(2) は T の定義から明らかである。($|T \cap \Sigma^2| \leq 1$ に注意。)

次に, 2進数を3進数に変換する関数 φ も n の領域を必要

とすることを示す。 $\Delta = \{0, 1, 2\}$ とおき, 任意の $w \in$

Δ^* に対して $[w]_3$ を w に対応する3進数とする。関数 φ :

$\Sigma^* \cdot 1 \rightarrow \Delta^* \cdot \{1, 2\}$ を次のように定める。

$$\varphi(x) = y \iff [x] = [y]_3$$

定理 4.3 関数 φ は, n の領域を必要とする。

証明 $B = \{w \in \Delta^* \cdot \{1, 2\} \mid [w]_3 = 3^k, k \in \mathbb{N}\}$

とおくと, B は正規集合 O^*1 であるから $O(1)$ 領域認識可能である (補題 2.1)。 T は φ によつて B に還元可能である。よつて, 定理 4.1 と補題 3.2 より, この定理が証明される。 \square

5. 整数の乗算

本節では, 整数の乗算が n の領域を必要とすることを示す。

記法 2 節で Σ^+ の元 w に対して, w に対応する整数を示す

$[w]$ という記法を定めた。逆に, 正整数 n に対して, n に 2 進数列 $w (\in \Sigma^* \cdot 1)$ (但し, 左側が低位, 右側が高位で, 最高位ビットの最右端は 1) を $[w]$ と記す。すなわち,

$[n] = a_1 a_2 \dots a_k (a_k = 1)$ ならば,

$$n = \sum_{i=1}^k a_i \cdot 2^{i-1}$$

である。便宜上, $[0] = 0$ とおく。任意の自然数 n に対して, $[([n])] = n$ が成り立つ。

また, $\bar{\Sigma} = \Sigma \cup \{\phi\}$ ($\phi \notin \Sigma$) とおく。 Σ^* の任意の 2 元 $x = a_1 a_2 \dots a_m, y = b_1 b_2 \dots b_n$ に対して,

$$x \# y = a_1 b_1 a_2 b_2 \dots a_l b_l$$

と定める。ただし, $l = \max\{m, n\}$ であり, $m < n$ なら

ば $a_{m+1} = \dots = a_e = \phi$, $m > n$ ならば $b_{n+1} = \dots = b_e = \phi$ である。以上の準備のもとに乗算 $MULT: \Sigma^* \rightarrow \Sigma^*$ を、次のように定める。

$$MULT(w) = \begin{cases} J([X] * [Y])I & \text{ある } x, y \in \Sigma^* \text{ に対して,} \\ & w = x \# y \text{ となるとき} \\ \text{定義されない} & \text{それ以外のとき} \end{cases}$$

ここで、 $*$ は通常の整数の乗算を表わす。すなわち、 $MULT$ は2つの2進数を各々下位の桁から読み、それらの積を下位の桁から出力する自然な乗算である。

定理 5.1 乗算 $MULT$ は、 n の領域を必要とする。

この定理の証明のため、平方数を与える関数 $SQR: \Sigma^* \rightarrow \Sigma^*$ を次のように定義する。任意の $w \in \Sigma^*$ に対して、

$$SQR(w) = J([w]^2)I.$$

すなわち、 $[SQR(w)] = [w]^2$.

補題 5.2 関数 $S: \mathbb{N} \rightarrow \mathbb{N}$ に対して、 $MULT$ が $S(n)$ 領域計算可能であることと、 SQR が $S(n)$ 領域計算可能であることは同値である。

証明 $MULT$ が $S(n)$ 領域計算可能であれば、 SQR もそうであることは明らか。逆に、 SQR が $S(n)$ 領域計算可能であるとする。任意の整数 a, b に対して、

$$a * b = \{(a+b)^2 - (a-b)^2\} / 4$$

が成り立つことと、加算、減算が $O(1)$ 領域で計算可能であることから、MULT も $S(n)$ 領域計算可能となる。□

定理 5.1 を証明するためには、次の定理を示せばよい。

定理 5.3 SQR は n の領域を必要とする。

この定理は、次のいくつかの補題を用いて証明される。まず、集合 $A, B (\subseteq \Sigma^*)$ を次のように定義する。

$$A = \{ w \in \Sigma^* \mid w = 1^{e_1} 0^{f_1} 1^{e_2} 0^{f_2} \dots 0^{f_t} 1^{e_{t+1}}$$

($e_1, \dots, e_{t+1}, f_1, \dots, f_t > 0, t \geq 3$) と表

わされ、 f_1, f_2, \dots, f_t の中で 3 番目に大きな

ものの 1 を $f_s (1 \leq s \leq t)$ とすると、部分列

$1^{e_{s+2}} 0^{f_{s+2}} \dots 0^{f_t} 1^{e_{t+1}}$ の左端から右に f_{s+1} 番目

の文字は 1 である }。

$$B = \{ w \in \Sigma^* \mid w = 1 w' 1 \text{ か } SQR(w) \in A \}.$$

補題 5.4 A は $\log n$ 領域認識可能である。

略証 3本の補助テープを用いて、それまでに入力された(両端を1ではさまれた)0だけから成る部分列のうち、最も長い列、2番目に長い列、3番目に長い列のそれぞれの長さとしてそれに対応する“ f_{s+1} 番目のビット”の内容とを記憶しておけばよい。□

補題 5.5 B は n の領域を必要とする。

この証明も補題 3.1 をもとにしてなされる。任意の $n \in$

N に対して,

$$B^{(n)} = \{u \mid u \in \{1, \Sigma^*\}, |u| = n, \text{ある } v \in \Sigma^* \text{ に対して } uv \in B\}.$$

とおく。次の2つの補題を示せばよい。

補題5.6 任意の $n \geq 2$ に対して, $|B^{(n)}| = 2^{n-2}$.

補題5.7 $n > 2$ とする。 $B^{(n)}$ の中の任意の相異なる2元

u_1, u_2 に対して, $v \in \Sigma^*$ が存在して,

$$u_1 v \in B \text{ かつ } u_2 v \notin B$$

$$\text{または } u_1 v \notin B \text{ かつ } u_2 v \in B$$

となる。

定理5.3の証明 SQRが,

$$\inf_{n \rightarrow \infty} \frac{S(n)}{n} = 0$$

を満たす S について, $S(n)$ 領域計算可能であるとする。すると, B の認識がやはり $S(n) + \log n$ 領域でできてしまうことが次のようにしてわかる。

入力列 w に対して, w を2乗する。これは SQR によつて $S(n)$ の領域でできる。次に, 得られる列 u ($[u] = [w]^2$) が A に属するかどうかを調べる。これは, $\log n$ の領域でできる(補題5.4)。従つて, B は $S(n) + \log n$ 領域認識可能となり, 補題5.5に反する。

すなわち, SQR も n の領域を必要とする。 □

以上で、最初に定めた乗算 MULT が n の領域を必要とする
ことの証明が終わったが、2つ注意をしておく。

1. MULT は、 n の領域で計算可能である。
2. 以上の結果は、乗算 (MULT) の定義のしかたに特別に依
存するものではない。乗算の定義を別のものに変えても、“自
然”なものである限り、同様の結果が得られる。

6. オンライン計算に関する2つの性質

集合 $C (\subseteq \Sigma^*)$ を次のように定める。

$$C = \{ w \in \Sigma^* \mid w = u \cdot 1 \cdot 0^i \cdot 1 \text{ (} |u| \geq i, i \in \mathbb{N} \text{)} \text{ と表わ}$$

され、 u の右から i 番目のビットは 1 で
ある }.

C について、次のような性質が成り立つ。

- 補題 6.1
- (i) C は n の領域を必要とする。
 - (ii) C の逆語の全体 C^R は $\log n$ 領域認識可能であ
る。
 - (iii) C は非決定性 (オンライン) チューリング
機械を用いると、 $\log n$ 領域で認識可能であ
る。

この補題から次の2つの定理が得られる。

定理 6.2 $\log n$ 領域で認識可能な言語のクラスは、逆語

をとる操作に関して閉じていない。

決定性(非決定性)チューリング機械により $\log n$ 領域で認識可能な言語のクラスを ON-LINE DLOG(NLOG) SPACE と表わす。

定理 6.3 ON-LINE DLOG SPACE \subseteq ON-LINE NLOG SPACE

7. 算術式の変換

最後に、算術式を中置記法から後置記法(Polish 記法)に変換する問題を考える。

定義 \mathcal{J}, \mathcal{O} をアルファベットとし、 $\mathcal{J} \cap \mathcal{O} = \emptyset$ とする。

$\Gamma = \mathcal{J} \cup \mathcal{O}$, $\Delta = \Gamma \cup \{ (,) \}$ とおく。 Γ 上の 中置算術式 の集合 $AEXP (\subseteq \Delta^*)$ は、次の i), ii) を満たす最小の集合である。

i) $\mathcal{J} \subseteq AEXP$,

ii) $x, y \in AEXP$, $\sigma \in \mathcal{O}$ のとき $(x\sigma y) \in AEXP$.

Γ 上の 後置算術式 の集合 $POSTEXP (\subseteq \Gamma^*)$ は、次の i), ii) を満たす最小の集合である。

i) $\mathcal{J} \subseteq POSTEXP$,

ii) $x, y \in POSTEXP$, $\sigma \in \mathcal{O}$ のとき $x y \sigma \in POSTEXP$.

次の 3 つの条件を満足する関数 $\psi: \Delta^* \rightarrow \Gamma^*$ を算術式変換関数(または、目的コード生成関数)と呼ぶ。

i) $a \in \mathcal{J}$ に対して、 $\psi(a) = a$,

ii) $x, y \in \text{AEXP}$, $\sigma \in \mathcal{O}$ に対して, $\psi((x\sigma y)) = \psi(x)\psi(y)\sigma$,

iii) $x \notin \text{AEXP}$ ならば $\psi(x) \notin \text{POSTEXP}$.

定理 7.1 算術式変換関数 ψ は n の領域を必要とする。

この定理を示すために, 予備的考察を行なう。 $\Sigma = \{0, 1\}$ とする。

補題 7.2 i) 集合 $A = \{w \# w^{\#} \# \mid w \in \Sigma^*\}$ は n の領域を必要とする。

ii) 集合 $B = \{0^n 1^n \mid n \in \mathbb{N}\}$ は $\log n$ の領域を必要とする。

(i) は, Lewis, Stearns と Hartmanis [6] による。もちろん, 本稿の補題 3.1 を用いれば容易である。)

定理 7.3 POSTEXP は, $\log n$ 領域認識可能であり, また, $\log n$ 領域を必要とする (すなわち, $\log n$ の領域が必要かつ十分である)。

略証 $\log n$ 領域必要なことは次のように示される。補題 3.2 と補題 7.2 (ii) より, 集合 $\{a^n b^n \mid n \in \mathbb{N}\}$ が $\log n$ 領域を必要とすることが示される。 $a \in \mathcal{J}$, $b \in \mathcal{O}$ とすると,

$$\text{POSTEXP} \cap a^* b^* = \{a^n b^n \mid n \in \mathbb{N}\}$$

である。よって, 補題 2.1 と 2.2 より POSTEXP は $\log n$ の領域を必要とする。 \square

定理 7.4 AEXP は, n 領域認識可能であり, また, n 領域を必要とする (すなわち, n の領域が必要かつ十分である)。

略証 n 領域を必要とすることは、 $O(1)$ 領域計算可能な関数 f を構成することによって、補題 7.2(i) の集合 \mathcal{F} によって AEXP に還元可能であることを示すことにより証明される。 □

定理 7.1 の証明 AEXP は ψ によって POSTEXP に還元可能であるから、補題 3.2, 定理 7.3, 7.4 より明らか。

謝辞 本稿と関係なく日頃御世話頂く電気通信大学計算機科学科の諸氏、諸嬢、とくに西澤輝泰、樋川知子、竹元裕子各氏(ちなみに、竹元女史は本講究録中の“Lisp プログラム自動合成のシステム”の執筆者永井雅人氏の実姉である)に深甚の感謝の意を表したい。また、数ページにわたる難解な草稿を忍耐強く清書して頂いた同大学戸田誠之助氏に深く感謝する。

怨辞 本稿に関し、日頃有害無益な難癖をつけていただく相模工大岩田茂樹先生に怨念をこめて御礼を申し上げたい。

参考文献

1. Aho, A. V., J. E. Hopcroft and J. D. Ullman, The Design and Analysis of Computer Algorithms, Addison-Wesley, Reading, Mass., (1974).
2. Fischer, M. J., and L. J. Stockmeyer, Fast on-line integer multiplication, J. Comput. System Sci. 9, 317-331 (1974).
3. Gallaire, H., Recognition time of context-free languages by on-line Turing machines, Inform. Contr. 15, 288-295 (1969).
4. Hartmanis, J., and H. Shank, On the recognition of primes by automata, J. Assoc. Comput. Machinery 15, 382-389 (1968).
5. Hartmanis, J., and H. Shank, Two memory bounds for the recognition of primes by automata, Math. Systems Theory 3, 125-129 (1969).
6. Lewis, P. M. II, R. E. Stearns, and J. Hartmanis, Memory bounds for recognition of context-free and context-sensitive languages, IEEE Conference Record 6th Annual Symposium on Switching Circuit Theory and Logic Design, 191-202 (1965).
7. Paterson, M. S., M. J. Fischer, and A. R. Meyer, An improved overlap argument for on-line multiplication, in Complexity of Computation (SIAM-AMS Proceedings, Vol. 7), R. M. Karp, Ed., American Mathematical Society, Providence, Rhode Island, 97-111 (1974).