数理解析研究所講究録
992巻 1997年 138-145

チューリング機械の領域計算量の厳密な階層について

九州芸術工科大学　岩本 宙造　(Chuzo Iwamoto)
九州大学・工学部　岩間 一雄　(Kazuo Iwama)

1　はじめに

領域計算量を論じる際には，作業テープの記号数を固定するか否かで二つの場合が考えられる．記号数を固定しない場合には，$\inf_{n\to\infty}\frac{g(n)}{f(n)} = 0$ なる関数 $g(n), f(n)$ に対し，$f(n)$ 領域の決定性TMは $g(n)$ 領域の決定性TMより受理能力が真に高いことが知られている [1, 2]．線形加速定理が成立するので [1, 2]，この階層は稠密であると言える．一方，テープ記号を m に固定した場合には，それほど稠密な結果は知られていなかった．最も良い結果として，如何なる整数 $r \geq 1$，定数 $b > a \geq 0$ に対して，m 記号の bn^r 領域決定性TMは，m 記号の an^r 領域決定性TMより真に受理能力が高いことが知られていた [4]．（これは，如何なる $\varepsilon > 0$ に対しても，$(1+\varepsilon)an^r$ 領域は an^r 領域より受理能力が高いことを意味する．）テープ記号数を固定した場合には，$f(n)$ 領域の決定性TMは $(f(n)-c)$ 領域の決定性TMで模倣できることが知られているに過ぎないことから，[4] の階層には改善の余地が十分残されており，それほど厳密な階層とは言えない．

本稿では，作業テープの記号数を m に固定して，より厳密な領域計算量の階層を示す．$s(n)$ を任意の強構成可能関数とする．このとき，如何なる整数 m, 如何に小さい定数 $\varepsilon, \varepsilon_1, \varepsilon_2 > 0$ に対しても，以下の条件を満たす言語 $L \subseteq \{0,1\}^*$ が存在する．(i) m 記号の決定性TM T で $\left(s(n) + \frac{s(n)}{\sqrt{(\frac{2}{3}-\varepsilon)n}} + (4+\varepsilon_1)\log_m s(n) + (3+\varepsilon_2)\log_m n\right)$ 領域で受理できる．(ii) m 記号の如何なる決定性TM T_x でも $s(n)$ 領域では L を受理できない．(iii) T は必ず停止するが，T_x は無限ループに入ることで入力を受理しないこともある．$s(n)$ が $\log n$ より真に速く成長する関数とすれば，m 記号の $(1+o(1))s(n)$ 領域決定性TMは，m 記号の $s(n)$ 領域決定性TMより受理能力が高いと言える．つまり，[4] における定数係数 $1+\varepsilon$ は，「1に漸近する関数」に置き換えられたことになる．さらに，[4] では $s(n)$ は an^r と書ける関数と制限されていたが，我々の結果は，如何なる強構成可能関数 $s(n)$ に対しても成立する．

証明は，これまでに知られる階層の証明と同様に，対角線論法による．しかし，我々の結果は，これまでの文献にはなかった二つの新しいアイデアに基づいている．その一つは，入力テープの使い方である．我々のモデルは，いわゆるオフラインTMであり，二方向にヘッドを動かせる入力テープと作業テープからなる．階層定理の対角線論法による証明において重要な点は，T の入力テープ上に符号化列として与えられたTM T_x を，T で模倣する部分である．その際，T_x の符号化列の後ろに，十分長い挿入列 (padding sequence) を付け加えた列を，T の入力列として与えることで，T が T_x を模倣するのに十分な領域が確保されるようになる．そのため，挿入列は十分な長さがあればどのような列でも良く，1^l といった任意の列が使われてきた．本稿では，ある複雑な構造を持つ列を挿入列として選び，その構造を利用することで，より厳密な階層定理を証明する．直観的に言えば，それまでの挿入列では，入力列を利用して一つのカウンタを模倣できていたが，我々の新しい挿入列は，二つのカウンタを模倣することが可能となる．

もう一つのアイデアは，Sipserによって提案された手法の新たな応用である．階層定理の証明におけるもう一つの重要な点は，T_x が無限ループに入っているか否かを T が如何に判定するかである．最も単純な方法は，T_x の動作数を数え上げることであるが，T_x は $s(n)$ 領域なので，T は $s(n)$ 程度の余分な領域を必要とする．つまり，新たな $s(n)$ 領域をカウンタに使用し，残りの $s(n)$ 領域を T_x の作業テープの模倣に用いることになり，T が $2s(n)$ 領域TMになることは避けられそうにない．Sipserは，[7] において，受理計算状況から深さ優先で初期計算状況を探索する手法を提案している．この手法を使うことで，ごく少ない領域の増加で T_x がループに入るか否かを T は判定できるようになる．

2　これまでの結果とモデル

モデルは，いわゆる決定性オフライン チューリング機械 M であり，有限制御部，左右端に特別の終端記号の書かれた読みとり専用入力テープ，一本の作業テープから構成されている．ただし，作業テープに関しては，文献によって以下の点で定義が微妙に異なる．(i) 一方向に無限か，二方向に無限か，(ii) 作業テープ上に空白記号 B を書くことができるか否か，(iii) 一方向無限であれば，テープの他方の端には特別の終端記号が置かれているか否か．

テープ記号が制限されていない場合について階層定理を論じる場合は，これらの違いは問題は生じない．しかし，作業テープが一方向無限で，テープ記号は $0, 1, B$ のみ，左端の終端記号は使えないと仮定すれば，次のような問題が生じる．明らかに，M が最初にすべきこ

とは，特別な文字列（例えば，111）を作業テープの左端に書いて，ヘッドが作業テープから落ちないようにすることである．しかしながら，この方法では，左端の111を見分けるために，作業テープ上の他の場所に文字列111が現れないようにしなくてはならない．そうすると，必要な領域量が定数係数の範囲で増加してしまうことは避けられそうにない．もし，両方向に無限の作業テープを用いることが可能なら，空白記号Bを終端記号の代わりに使えるので，この問題は生じない．

テープ記号が制限された場合の階層定理に関しては，Ibarra [3]の結果が知られている．そのモデルは，一つの作業テープと一つの作業テープヘッドをもつ．Ibarraは，(i)一方向無限と二方向無限の両方の場合を考えており，(ii)作業テープのヘッドは空白記号Bを書けると仮定しているが，(iii)作業テープのヘッドが左端を認識できるとの仮定は使っていない．文献[3]において，$m+1$記号でn^r領域のTMで受理できるが，m記号でn^r領域の如何なるTMでも受理できない言語が存在することが示されている．この階層定理は，決定性と非決定性の両方の場合に成立する．しかし，作業テープが二方向無限で，かつ，決定性の場合は，テープ記号数$m+1$は$m+2$に増加する．

IbarraとSahni [4]は，[3]と同じモデルを用いて，一方向無限の作業テープの場合について，上記の階層を改善した．文献[4]において，如何なる整数$r \geq 1$，如何なる定数$b > a \geq 0$に対して，m記号でbn^r領域のTMで受理できるが，m記号でan^r領域の如何なるTMでも受理できない言語が存在することが示されている．

Seiferas [5, 6]は，領域やテープ記号，作業テープのヘッド数を制限することで，クラスを分離する多くの結果を示した．SeiferasのTMは，一つの作業テープ上に$l \geq 1$個のヘッドを持つ．Seiferasは，(i)作業テープは一方向無限，(ii)作業テープのヘッドは空白記号Bを書くことが可能で，(iii)作業テープのヘッドは左端を認識できると仮定している．文献[5]において，$s_0(n) - 2s(n) - l\log_m s(n) - \log_m n \neq O(1)$なる関数$s_0(n), s(n)$に対して，$m$記号で$s_0(n)$領域，$(l+2)$ヘッドの決定性TMで受理できるが，$m$記号で$s(n)$領域，$l$ヘッドの如何なる決定性TMでも受理できない言語が存在することが示されている．（この階層性は，作業テープのヘッド数を増加させてクラスの違いを示しているが，領域量に対数を加えることで作業テープのヘッド数を減らすことができることが知られている．つまり，作業テープの記号数がmに固定されている場合，$l+k$個のヘッドを持つ$s(n)$領域決定性TMは，（k個の新たなカウンタを使うことで）l個のヘッドを持つTMで$s(n)+(k+1+\varepsilon)\log_m s(n)$領域で模倣可能である．しかし，この模倣は$l=1$の場合は成立しない．）

3 結果

本稿のTMは，読みとり専用の入力テープ一本と読み書きできる作業テープ一本からなるオフラインTMである．入力テープの両端には，特別の終端記号が置かれている．本稿のTMは，入力テープと作業テープにそれぞれ一つのヘッドをもつ．また，(i)作業テープは二方向に無限で，(ii)作業テープのヘッドは，空白記号Bを書けないと仮定する．（従って，作業テープのヘッドが訪れたマスには必ず非空白記号が書き込まれている．）

最初，入力ヘッドは入力テープの左端のマスに置かれている．作業テープの記号は，$0, 1, \ldots, m-1$とする．最初，作業テープの各マスには空白記号Bが書かれている．初期状態，受理状態，非受理状態は，それぞれ唯一であり，q_1, q_2, q_3で表される．TMは，無限ループに入ることで入力列を受理しないこともある．全てのnに対して，如何なる入力が与えられても作業テープ上でちょうど$s(n)$マス使用するTMが存在するとき，$s(n)$を強構成可能関数という．

定理1. $s(n)$を任意の強構成可能関数とする．このとき，如何なる整数$m \geq 2$，如何に小さい定数$\varepsilon, \varepsilon_1, \varepsilon_2 > 0$に対しても，以下の条件を満たす言語$L \subseteq \{0,1\}^*$が存在する．(i) m記号の決定性TM Tで$\left(s(n) + \frac{s(n)}{\sqrt{(\frac{2}{3}-\varepsilon)n}} + (4+\varepsilon_1)\log_m s(n) + (3+\varepsilon_2)\log_m n\right)$領域で受理できる．(ii) m記号の如何なる決定性TM T_xでも$s(n)$領域では受理できない．(iii) Tは必ず停止するが，T_xは停止するとは限らない．（証明は節4で与える．）

系1. 強構成可能関数$s(n)$が，$s(n) \neq O(\log n)$を満たせば，m記号で$(1+o(1))s(n)$領域の決定性TMは，m記号の$s(n)$領域の決定性TMより受理能力が高い．

注意1. $s(n) \leq \sqrt{(\frac{2}{3}-\varepsilon)n}$のときは，定理1の$\frac{s(n)}{\sqrt{(\frac{2}{3}-\varepsilon)n}} + \log_m n$の項を除くことができる．（証明に幾つかの修正が必要である．詳しくは，節4で述べる．）故に，$s(n)$が$\log n$の多項式のときは，領域$s(n)$に$(2+\varepsilon_2)\log_m n$を加えるだけで，$m$記号の領域限定決定性TMで受理できる言語のクラスは真に大きくなることが分かる．

強構成可能関数には，整数$c \geq 1$，有理数$k \geq 1, t > 1$に対し，$n!, 2^n, n^k, \log^t n, c$などがある．また，$s_1(n)$と$s_2(n)$が強構成可能であれば，$s_1(n)+s_2(n), s_1(n)s_2(n), 2^{s_1(n)}$なども強構成可能である．

本稿では，二方向無限の作業テープを仮定し，空白記号Bは書けないものとしている．この仮定は，定理1

が全ての強構成可能関数に対して成立するという目的のために導入されている．しかし，$n!, 2^n, n^k, \log^t n, c$ など，良い振舞いをする具体的な関数や，これら特定の関数から構成される関数 $s_1(n)+s_2(n), s_1(n)s_2(n), 2^{s_1(n)}$ に対しては，定理1は[4]と全く同じモデル上で成立する．

注　意2. $s(n)$ を，$n!, 2^n, n^k, \log^t n, c$ といった良い振舞いをする具体的な関数，または，良い振舞いをする関数から構成される関数 $s_1(n) + s_2(n), s_1(n)s_2(n), 2^{s_1(n)}$ とする．このとき，定理1は，一方向無限の作業テープをもち，左端に特別の記号を持たない決定性TMで成立する．さらに，作業テープに空白記号 B を書けるとし，テープ記号を $B, 0$ のみに限定しても定理1は成立する．

故に，我々の結果は，全く同じモデル上で[4]の結果を改善するとともに，領域関数 $s(n)$ を一般化したことになる．[5, 6]のように特別の終端記号が作業テープの一端に置かれているとすると，我々の階層定理は一方向無限の決定性TM上で全ての強構成可能関数 $s(n)$ に対して成立する．

4　定理1の証明

如何なる $s(n)$ 領域TM T_x でも受理できない言語 $L \subseteq \{0,1\}^*$ を，$\left(s(n)+\frac{s(n)}{\sqrt{(\frac{2}{3}-\varepsilon)n}}+(4+\varepsilon_1)\log_m s(n)+(3+\varepsilon_2)\log_m n\right)$ 領域で受理するTMで，必ず停止する T を作成する．まず最初に，TMの符号化規則を定める．以下では，$s(n) \geq \sqrt{(\frac{2}{3}-\varepsilon)n}$ と仮定する．（$s(n) < \sqrt{(\frac{2}{3}-\varepsilon)n}$ の場合は，幾つかの修正が必要であり，必要に応じて以下の各節で説明する．この修正により，定理1から $\frac{s(n)}{\sqrt{(\frac{2}{3}-\varepsilon)n}} + \log_m n$ の項を除くことができる．）

4.1　TMの符号化規則

TMの状態を $q_1, q_2, \ldots$ で表す．状態 q_i は，長さ $i+1$ の文字列 10^i に符号化され，作業テープの記号 $B, 0, 1, 2, \ldots$ は，文字列 $10, 100, 1000, 10000, \ldots$ に符号化される．文字列10　(100) はヘッドが左（右）動くことを意味する．例えば，遷移関数 $\delta_1(q_5, B, 2) = (q_3, 1, \mathrm{L}, \mathrm{R})$ は，以下の文字列に符号化される．

$$\underbrace{100000}_{q_5}\ \underbrace{10}_{B}\ \underbrace{10000}_{2}\ \underbrace{1000}_{q_3}\ \underbrace{1000}_{1}\ \underbrace{10}_{\mathrm{L}}\ \underbrace{100}_{\mathrm{R}}$$

TMの符号化は，遷移関数の符号化を連結してできる文字列の後ろに初期状態，受理状態，非受理状態を表す文字列を連結したものである．さらに，このTMの符号化の後ろには，任意に長い文字列が連結されている．TMの符号化の部分を符号化部 (encoding part)，その後ろの任意に長い文字列を挿入部 (padding part) と呼ぶ．挿入部は，以下の構造を持つ（図1も参照）．

$$\underbrace{110^b10^1}\,\underbrace{110^b10^2}\cdots\underbrace{110^b10^h}\cdots\underbrace{110^b10^b}\,11\,\underbrace{00\cdots0}_{\leq 3b+2} \tag{1}$$

h 番目の部分列は，長さ $b+3$ の文字列 110^b1 と長さ h の文字列 $00\cdots0$ からなる．一番最後の部分列は 110^b10^b であり，その後ろには長さが高々 $3b+2$ の文字列 $00\cdots0$ が連結されている．挿入部の最後の文字列 $00\cdots0$ の長さが $3b+2$ より大きければ，$b+1$ を改めて b とおいて上記と同じ構造が作れてしまうことに注意が必要である．

如何に成長が遅い関数 $\psi(n) \neq O(1)$ に対しても，十分長い挿入部が符号化部に連結されていれば，入力列の先頭から $\psi(n)$ 番目までに符号化部が含まれるようになる．そのような十分長い挿入部には，如何なる小さい $\varepsilon > 0$ に対しても $b \geq \sqrt{(\frac{2}{3}-\varepsilon)n}$ を満たす部分列 110^b10^b が含まれることになる．

T は，その入力列 x を，TM T_x の符号化部と挿入部の連結とみなし，T が x を受理するのは，T_x が x を受理しないときに限るとする．言い換えれば，T がその入力 x を受理しないのは，以下の条件が一つでも満たされない場合である．(i) x が文法的に間違っている．(ii) x の符号化部がいずれかの決定性TMに対応しない．(iii) x の挿入部の構造が正しくない．

4.2　T_x の計算状況の表現

T は，作業テープを11個のブロックに分ける．これらは，ブロック $1, 2, \ldots, 9$，ブロック A, B と呼ばれる（図1参照）．二つのブロック間の境界は長さ t の文字列 1^t である．ただし，$t \geq 3$ は十分大きい定数である．（t の値は，後で与える不等式(2)が成立するように大きくとる．）ブロック1,4,6は，それぞれ T_x の状態，入力ヘッド位置，作業テープのヘッド位置を表すために用いる．T_x は，m 個の記号が使える．そこで，ブロック3は m 進カウンタ，ブロック7は T_x の作業テープ上で使用したマスの数を m 進数の文字列として蓄える．さらに，ブロック5,8は b と $s(n)$ の値を m 進数の文字列としてそれぞれ蓄える．（ブロック3,5,7,8の使用目的は後で述べる．）T_x の作業テープはブロック9で表されるが，extra bitと呼ばれる記号が b 文字ごとに等間隔で挿入されている（図1参照）．extra bitは0または1である．（b の値は，入力の挿入部の b と同じであることに注意．）二つのextra bitに挟まれた長さ b の領域をサブブロックと呼ぶ．extra bitは，本証明で重要な役割を演じる．この点については後で述べる．各extra bitは，(i) それに引き続くサブブ

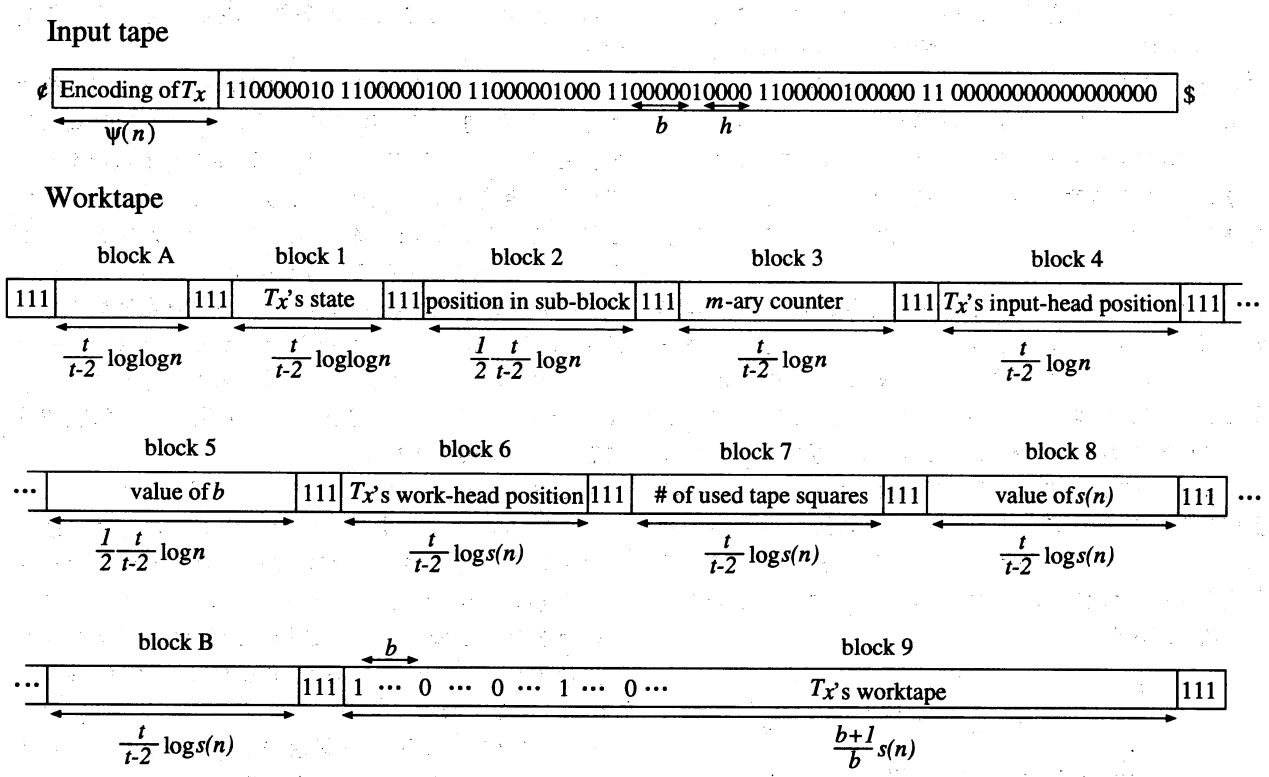

Figure 1: Input tape and worktape of T

ロックにT_xの作業テープのヘッドが存在するか否か，(ii) どれが最初のサブブロックで，どれが最後のサブブロックかを表す．図1では，4番目のサブブロックの直前のextra bitが1となっている．これは，T_xの作業テープのヘッドが4番目のサブブロックのあるマスに置かれていることを意味する．最初と最後を除く残り全てのextra bitは0である．($s(n) < b$のときは，ブロック9にはextra bitは挿入されない．ブロック9は長さ$s(n) < b$の一つのサブブロックから構成されていると考える．) T_xの作業テープのヘッドが，あるサブブロックの中のh番目のマスに置いてあるとき，ブロック2にはhの値が蓄えられている．

ブロック9には，extra bitがb文字ごとに等間隔で挿入されているので，長さは$\frac{b+1}{b}s(n)$となる．(bの値は，入力列の挿入部に与えられている．) ($s(n) < b$のときは，ブロック9にはextra bitは挿入されないので，長さは$s(n)$のままである．また，ブロック5も不要となる．さらに，ブロック2はブロック6で代用できる．したがって，定理1から$\frac{s(n)}{\sqrt{(\frac{2}{3}-\varepsilon)n}} + \log_m n$の項を除くことができる．) もし，ブロックの中に文字列1^tが現れるとすると，Tはその文字列1^tと境界を表す文字列1^tの区別がつかなくなる．そこで，長さ2の文字列00を$t-2$マスごとに等間隔で挿入する．故に，ブロックA,1，ブロック2,5，ブロック3,4，ブロック6,7,8,Bの長さは，それぞれ，$\frac{t}{t-2}\log_m \log_m n$，$\frac{1}{2}\frac{t}{t-2}\log_m n$，$\frac{t}{t-2}\log_m n$，$\frac{t}{t-2}\log_m s(n)$となる．(こうすることで，挿入された空間00の上に必要に応じて目印01を置くことが可能となる．以下では，二つのブロックに含まれる値が等しいかを判定する場面が頻繁に出てくる．そのような場合，長さ$t-2$の各文字列を対応する同じ長さの文字列と比較していく．その際，今どの文字列とどの文字列を比較しているのかを示すために，二つの目印01を用いる．)

$s(n)$は，強構成可能関数であるので，まずTは作業テープ上でちょうど$s(n)$マス使うTMを模倣する．Tは，全ての非空白記号を0に変え，その両端に境界を表す1^tを生成する．(0が書かれた部分がブロック9になるが，extra bitは後で挿入する．) これらの0の数をm進カウンタで数え上げることにより，TはブロックBを生成することができる．ブロック6,7,8も同様である．ブロック3,4は，入力列xの長さを数え上げることにより生成できる．ブロック2,5は，入力列xの長さを一文字置きに数え上げることにより生成できる．また，ブロック1,Aはブロック3の長さ$\log_m n$を数え上げることにより生成できる．今のところ，ブロック9は$s(n)$マスしか含んでいない．そこで，以下の方法で記号0をb文字ごとに等間隔で挿入することで長さを$\frac{b+1}{b}s(n)$にする．(i) Tは，入力ヘッドを長さbの文字列$00\cdots0$の最初の文字の上におく．(ii) Tは，その作業ヘッドをブロック9の最初の文字の上に置く．Tの入力ヘッドがh個の0を順に読んでいく作業と並行して，Tの作業テープのヘッドを右へ動かしていく．

Tの入力ヘッドが記号1に着いたとき，Tの作業テープのヘッドはブロック9のb番目のマスに着いたことになる．(iii)そこで，一つの記号1が挿入される．つまり，ブロック9の右端の境界を右に一マスシフトする．この手続きを繰り返すことで，Tはブロック9に含まれるマスの数を$\frac{b+1}{b}s(n)$に増やすことができる．最後に，ブロック9に含まれる文字1の全てを0に変える．

ひとたび上記11ブロックが生成されたら，Tは境界記号1^tを決して消したり，移動したりしない．さらに，Tはブロック9に含まれるextra bitの位置も変えない．Tの作業テープのヘッドは，常に11ブロックの中で作業を行なう．もし，TがT_xを模倣するのにより多くの領域が必要であることが判明したら，Tは非受理状態で停止する．したがって，如何に小さい定数$\varepsilon, \varepsilon_1, \varepsilon_2 > 0$が与えられても，大きな$n$に対して，$T$が使うマスの数が以下の式で押えられるような十分大きい定数tが存在する．

$$\begin{aligned} & \frac{1+b}{b}s(n) + \frac{4t}{t-2}\log_m s(n) + \frac{3t}{t-2}\log_m n \\ & + \frac{2t}{t-2}\log_m \log_m n + 12t \\ \leq\ & s(n) + \frac{s(n)}{\sqrt{(\frac{2}{3}-\varepsilon)n}} + (4+\varepsilon_1)\log_m s(n) \\ & + (3+\varepsilon_2)\log_m n \end{aligned} \tag{2}$$

4.3 T_xの動作の模倣

まず，Tは次の条件を満たすTM（T_xとする）が存在するかを判定する．(i)入力列xの符号化部がT_xの正しい符号化になっている．(ii)T_xは決定性である．(iii)符号化部は(1)で示した構造を持つ．(i)は，有限オートマトンのように単純に前から読んでいくだけで判定できる．(ii)Tは，まずT_xの符号化をブロック1にコピーする．（入力列xの符号化部が十分長ければ，入力列の先頭から$\psi(n) \leq \log_m \log_m n$までに$x$の符号化部が含まれる．そうでなければ（ブロック1に入りきらなければ）Tは非受理状態で停止する．）入力ヘッドと作業テープのヘッドを用いて，遷移関数の全ての組を比較することで，TはT_xが決定性か否かを判定する．(iii)Tは，挿入部に含まれる11なる文字列の後ろに同じ長さの文字列$00\cdots0$を従えているかを調べる．この$00\cdots0$の長さをbとおく．m進カウンタを用いて，符号化部の最後の0の数が高々$3b+2$個であることを確かめる．各h対して，Tは二つの隣接する部分列中の10^hと10^{h+1}をブロック3のm進カウンタを使って比較する．さらに，Tは最後の部分列が110^b10^bであることを確かめる．もし，xが条件(i), (ii), (iii)を満たさない場合は，Tは非受理状態で停止する．以下では，入力列xはこれらの条件を満たしていると仮定する．

T_xの1ステップの動作を，以下の7ステップで模倣する．(i)最初，Tの作業テープはT_xの初期計算状況を含んでいる．(ii)Tは，T_xの現在の計算状況を含んでいると仮定する．(iii)TはT_xの入力ヘッドが今何を読んでいるかを調べる．(iv)TはT_xの作業テープのヘッドが今何を読んでいるかを調べる．(v)TはT_xの現在の状態を調べる．(vi)Tは，状態遷移関数の中でT_xの状態，入力ヘッドが読んでいる文字，作業テープのヘッドが読んでいる文字が一致するものを見つける．(vii)TはT_xの計算状況を，(vi)で見つけた状態遷移関数に従って遷移させる．

(i)最初，T_xの作業テープのヘッドは最初のマスに置かれているので（つまり，最初のサブブロックの最初のマスの上に置かれているので），最初のサブブロックの直前のextra bitは1となっている（サブブロックの境界は1となる）．（最初と最後のサブブロックは，他のサブブロックと区別するため，直前のextra bitが常に1となっている．したがって，もしT_xの作業テープのヘッドが最初か最後のサブブロックにあるときは，Tはそのことを状態で記憶しておく．）ブロック9の残りのマスは，全て0である．また，T_xの作業テープのヘッドは，サブブロックの最初のマスに置いてあるので，ブロック2は値1をm進数として蓄える．ブロック1はT_xの初期状態q_1を蓄える．ブロック3は，m進カウンタであり，最初は値0が入っている．入力ヘッドは最初入力テープの最初のマスに置かれているので，ブロック4は値1を含んでいる．ブロック6,7は値0が蓄えられている．ブロック5,8は，それぞれ常に値$b, s(n)$が蓄えられている．厳密に言えば，ブロック9の全てのマスは最初は空白記号Bでなくてはならない．その代わりに，ブロック9の全てのマスにはBの代わりに0が入っている．そこで，ブロック6,7を使った特別な手続きで，T_xの作業テープのヘッドがBを読んでいるか否かを判定するようにしている．（この手続きに関しては後述する．）

(ii)ブロック1～9は，T_xの現在の計算状況を表しているとする．

(iii)ブロック4に含まれる値をiとおく．Tの作業テープのヘッドは，値0をブロック3にm進数として書く．また，Tの入力ヘッドは入力列xの左端の記号の上に置く．(a)Tの入力ヘッドを右方向へ一マス動かし，(b)Tの作業テープのヘッドはブロック3の値を1だけ増加させる．ブロック3,4に含まれる値が一致するまで，(a), (b)の手続きを繰り返すことにより，Tの入力ヘッドを入力列xのi番目の記号の上に動かすことができる．

(iv)ブロック9では，最初は全ての記号が0であっ

たことから分かるように，ブロック9には，T_xの作業テープがBを読んでいるか否かに関する情報は何も含まれていない．そこで，Bを読んでいるか否かについては，ブロック6,7を使って判断する．もしブロック6,7の値が等しい，または，もしブロック6の値が-1ならは，TはT_xの作業テープのヘッドは空白記号Bを現在読んでいると判断できる．ブロック6,7はこの目的のためのみに用いる．（T_xの作業テープが読んでいる文字を見つける際には，Tはブロック2，extra bit，入力列xの挿入部を使う．）

T_xの作業ヘッドは今Bを読んでいないとする．仮定よりブロック9,6は，それぞれT_xの作業テープ，T_xの作業テープのヘッド位置を蓄えている．ブロック6に蓄えられている値をjとおく．まず，以下のような単純な手続きを考えてみる．(a) (iii)と同様の方法で，Tの入力ヘッドを入力列xのj番目の文字に置く．（つまり，Tは入力ヘッドを右に動かしていくのと並行して，ブロック3の値をブロック5の値に一致するまで0から一つずつ大きくしていく．）(b) Tの作業テープをブロック9の最初のマスへ動かす．(c) Tの入力ヘッドをj番目のマスから左へ動かすのと並行して，Tの作業ヘッドをブロック9の最初のマスから右へ動かしていく．Tの入力ヘッドが入力テープの左の終端記号に着いたとき，Tの作業テープはブロック9のj番目のセルに着く．（つまり，入力列を一進カウンタとして使う．）しかしながら，この方法では，次の二つの重大な問題が生じる．(1) $j \leq n$の場合しか成り立たない．(2) 入力ヘッドは，今，入力テープの左端のマスに置かれているので，Tの入力テープはjの値を再びカウントすることができない．したがって，この方法ではTの作業テープのヘッドは，ブロック9の最初のマスに戻ることができない．（もし，作業テープが一方向無限で，しかも，特別の終端記号が作業テープの左端に置かれているとすれば，Tの作業テープは作業テープの左端にヘッドを移動してから，ブロック9の先頭にヘッドを移動できる．二方向無限で空白記号Bが書けないとする我々のモデルの場合を考える．もし，空白記号Bを終端記号の代わりに用いたとすれば，そのBは非空白記号に変えられるので，使用するマスの数が増加してしまう．）そこで，入力列の挿入部をうまく利用する．

仮定より，(a) ブロック9はT_xの現在の作業テープの内容を蓄えており，(b) ブロック9ではb文字ごとに等間隔でextra bitが挿入され（各b文字はサブブロックと呼ばれる），(c) サブブロックの直前のextra bitが1であるのは，それが最初または最後のサブブロックであるとき，または，そのサブブロックにT_xの作業テープのヘッドが存在するときであり，(d) T_xの作業テープのヘッドのサブブロックの中での位置はブロック2に蓄えられている．

入力列の挿入部は，(1)に示す構造となっており，110^bなる部分列でh番目のものは，直後に10^hが続いている（図1も参照）．ブロック2に含まれる値をhとおく．(iii)と同様にして，（h個の11を通り越して）Tの入力ヘッドをh番目の部分列110^b10^hへ持っていく．ここでは，前半部分0^bと後半部分0^hを，それぞれサイズb,hの一進カウンタとして用いる．これら二つをそれぞれカウンタU_b, U_hと呼ぶ．まず，手続きの概略を示しておく．(a) TはカウンタU_hを見つける．(b) カウンタU_hに隣接するカウンタU_bを使って，Tの作業テープのヘッドをextra bit 1へ移動させる．（extra bit 1は，その直後のサブブロックにT_xの作業テープヘッドが含まれることを表す．）(c) カウンタU_hを使って，そのサブブロック中のh番目のマスにTの作業ヘッドを動かす．（$s(n) < b$のときは，block 9は長さ$s(n) < b$のサブブロック一つから構成される．したがって，手続きのうちU_bを使う部分は不要である．）

Tの作業テープのヘッドは，最初のサブブロックの直前のextra bitの上に置かれているとする．Tの入力ヘッドをU_b上を左から右へ動かすのと並行して，Tの作業テープのヘッドを右へ動かす．こうすることで，Tの作業テープのヘッドは2番目のextra bitにたどりつく．次に，Tの入力ヘッドをU_b上を右から左へ動かすのと並行して，Tの作業テープのヘッドを右へ動かす．こうして，Tの作業テープのヘッドは3番目のextra bitにたどりつく．Tの作業テープのヘッドがextra bit 1と出会うまで，この手続きを繰り返す．（extra bit 1と出会えば，その直後のサブブロックにはT_xの作業テープのヘッドが含まれる．）

次に，Tの入力ヘッドをカウンタU_h上で左から右へ動かすのと並行して，Tの作業テープのヘッドを右へ動かす．こうすることで，Tの作業テープのヘッドは，そのサブブロックのh番目のマスにたどりつく．こうして，Tは，T_xが今何を読んでいるかを知ることができる．そして，Tの入力ヘッドをU_h上を左から右へ動かすのと並行して，Tの作業テープのヘッドを左へ動かす．こうして，Tの作業テープのヘッドをextra bitの上に戻すことができる．上記と同様に，カウンタU_bを繰り返し用いることにより，Tの作業テープのヘッドを最初のextra bitの上に戻すことができる．（最初のextra bitは1なので，他のextra bitとは見わけがつく．）

(v) T_xの現在の状態はブロック1に蓄えられている．Tの作業テープのヘッドを単にブロック1に移動させるだけでよい．

(vi) Tの入力ヘッドは，入力列xの符号化部の状態遷移関数を順に見ていき，T_xの現在の計算状況に一致する状態遷移関数（δ_1とする）を見つけ出す（後述の

図3参照）．

(vii) Tはブロック1に蓄えられたT_xの状態を遷移関数δ_1に従って変化させる．もしT_xの入力ヘッドが右（左）に動くとき，Tはブロック4に蓄えられた値に1を加える（値から1を引く）．ここで，三つの場合が考えられる．(a) ブロック6,7に蓄えられた値が等しければ（T_xの作業テープのヘッドが右端の非空白記号の隣のBを読んでいれば），または，ブロック6の値が-1のときは（T_xの作業テープのヘッドが左端の非空白記号の隣のBを読んでいれば），ブロック7の値を1だけ増加させる．ブロック7の値がブロック8の値より大きければ，Tは非受理状態で停止する．そうでなければ(b)を実行する．(c) (iv)と同様にして，Tの作業テープのヘッドはextra bitが1であるサブブロックのh番目のマスにアクセスしてδ_1に従って記号を書き換える．（ただし，ブロック6の値が$h=-1$だったときは後で述べる処理をする．）

T_xの作業ヘッドが右（左）へ動くときは，Tはブロック2の値を1だけ大きく（小さく）する．もし，ブロック2の値がbで，かつ，T_xの作業テープのヘッドが右へ動く時はブロック2の値を0にし，1が立っていたextra bitを0に変え，その次に現れるextra bitを1にする．もし，ブロック2の値が1で，かつ，T_xの作業テープのヘッドが左へ動く時はブロック2の値を0にし，1が立っていたextra bitを0に変え，その前のextra bitを1にする．

もしブロック6の値が$h=-1$であった（つまり，今は-2または0に変えられている）場合は，ブロック6の値に1を加え，extra bitを除くブロック9の内容を全体に渡って右に1マスだけシフトする．（extra bitの位置はTは絶対に動かさないことに注意．）そうするために，Tの入力ヘッドを最初のU_bの先頭に持っていく．Tの作業テープがブロック9の内容をシフトしている間，Tの作業テープのサブブロック内でのヘッド位置を，U_b内での入力ヘッド位置で模倣する．こうすることでTの作業テープはextra bitを認識することができる．また，最初と最後のextra bitは1が立っているのでブロック9からTの作業テープのヘッドが飛び出すことはない．シフトによって空けられた最初のマス（最初のサブブロックの最初のマス）に，δ_1に従って記号を書き加える．

4.4 T_xがxを受理するか否かの判定

いま，T_xがxを受理しないときに限り，Tは入力列xを受理するというTを作成している．ここで注意すべきこととして，T_xは無限ループに入ることでxを受理しないこともあり得るということである．もし，TがT_xを単純に初期状況から最終状況に向けて模倣していくと，TはT_xがループに入っているのか否かが認識できなくなる．（つまり，T_xが無限ループに入ってxを受理しないときは，Tも無限ループに入ってxを受理しなくなる．）Sipser [7]は，T_xの動作数を数え上げることなしに，この判定が可能であることを示した．ここでは，その手法を適用する．

以下では，TM T_xが入力を受理するときは，全ての非空白記号を0に変え，左端の0の隣のBにヘッドを戻し，唯一の受理状態で停止すると仮定する．（この仮定で，問題は何も生じないということを補題1で述べる．）こうすることで，$s(n)$領域限定における異なる受理計算状況の数は$s(n)$となる（つまり，最終状況における作業テープの使用量の違いだけである）．作業テープ上にt個の0を含む受理計算状況をa_tで表す．ただし，$t=1,2,3,\ldots,s(n)$とする．

各節点が$s(n)$領域限定の計算状況を表し，枝が計算状況の間の遷移を表した有限の有向グラフを考える．T_xは決定性であるので，このグラフの出次数は高々1である．つまり，a_tは木の根になるということである（図2参照）．T_xが入力列xを受理するのは，T_xの初期状況が，ある受理状況a_tを根とする木に属している場合に限られる．$t=1,2,\ldots,s(n)$の各値に対して，Tはa_tを根とする木に深さ優先探索を実行する．

以下の記述は，[7]を元にしている．各tに対してa_tを根とする木の全ての節点を訪れるためには，Tは，(i) tの値，(ii) T_xの現在の計算状況C，(iii) 1ステップ前のT_xの計算状況（C'とする）を現在の計算状況Cに遷移させることの可能な遷移関数δ_iを憶えておかなくてはならない．ブロックBの長さは，$\frac{t}{t-2}\log_m s(n)$であるので（図1参照），Tはtの値をブロックBに蓄える．T_xの現在の計算状況Cは，ブロック1～9で表現する．挿入部が十分長い入力文字列xを考えれば，如何に成長の遅い関数$\psi(n)\neq O(1)$に対しても，先頭から$\psi(n)$番目までに符号化部が含まれることになる．そのような入力xに対しては，遷移関数δ_iは長さ$\log_m \log_m n$のブロックAに蓄えることができる．もし，ブロック6に蓄えられた値が，以下の手続き中に$s(n)$の値より大きくなれば，Tは非受理状態で停止する．

(a) T_xの現在の計算状況をCとおく（図2(a)参照）．節4.3と同様にして，T_xの作業テープのヘッドが今読んでいるマスと左右のマスに書かれた記号，T_xの入力ヘッドが今読んでいるマスと左右のマスに書かれた記号，T_xの今の状態 をTは知ることができる．T_xの符号化は遷移関数の符号化を連結したものであり，それらは入力列xの符号化部に含まれている（図3参照）．そこで，T_xの遷移関数を左から右へ順に見ていき，1ステップの遷移で計算状況C'をCに遷移できる遷移関数（δ_1とする）が存在するかを調べる．もし，そのようなδ_1が見つかれば，T_xの遷移関数をC'に戻

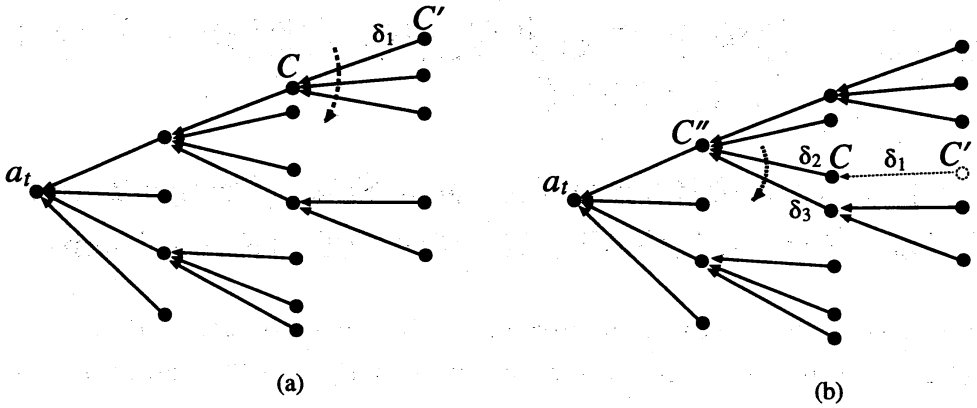

Figure 2: Depth-first search of the tree rooted at some a_t

す（図2(a)）．もし，C' が初期計算状況でなければ，T は再び手続き (a) を C' に対して適用する．

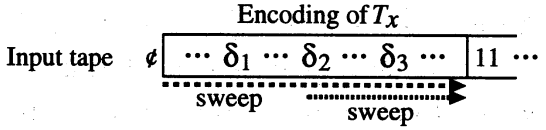

Figure 3: Sweep of the next move functions

(b) もし，そのような δ_1 が存在しない場合は（図2(b)参照），節4.3と同様に T は単純に T_x の動作を1ステップだけ模倣する．1ステップ後の計算状況を C'' とする．C を C'' に遷移させる遷移関数を δ_2 とする．次に，C'' に対して手続き (a) を適用するが，遷移関数を順に見ていく際には δ_2 から右の遷移関数のみを順に見ていく（図2(b), 3参照）．

T は $t = 1, 2, \ldots, s(n)$ の全てに対して，a_t を根とする木の全節点を訪れる．もし，ある a_t を根とする木の中に T_x の初期計算状況が見つかった場合は，T は非受理状態で停止する．もし，T_x の初期計算状況が，$t = 1, 2, \ldots, s(n)$ の如何なる t に対しても a_t を根とする木に含まれないときは，T は入力列 x を受理する．次の補題で本定理の証明を終る．

補題1. m 記号で $s(n)$ 領域の如何なるTMでも，上記のように定義したTM T で受理される言語を受理することはできない．

証明. T で受理される言語を受理できる m 記号，$s(n)$ 領域のTM（T_x とする）が存在すると仮定する．すると，次の条件 (i), (ii) を満足するTM T'_x が存在する．(i) T'_x は T_x と同じ言語を受理する．(ii) T'_x は入力を受理する際には，全ての非空白記号を0に変え，左端の0の隣の空白記号 B の上にヘッドを戻し，唯一の受理状態で停止する．TMが全ての非空白記号を0に変えるとき，作業テープのヘッドは非空白記号の左右端に隣接する二つの空白記号 B を0に変えることも考えられる．故に，T_x は $s(n)+2$ 領域TMである．如何なる $s(n)+2$ 領域TM T'_x に対しても，T'_x と同じ言語を受理する $s(n)$TM T''_x が存在する [4]．従って，上記条件を満たし，T と同じ言語を受理する $s(n)$ 領域TM T''_x が存在する．

T''_x の符号化に十分長い挿入部を連結した入力列 x を考える．この x を T に入力列として与えたら，T が使うマスの数は不等式 (2) で押えられる．故に，上記で述べた模倣が問題なく進み，T が x を受理するのは，T''_x が受理しないときに限られる．これは，仮定に矛盾する．□

参考文献

[1] J. Hartmanis, P.M. Lewis II and R.E. Stearns, Hierarchies of memory limited computations, in: *Proc. 6th IEEE Symp. on Switching Circuit Theory and Logical Design*, pp. 179-190, 1973.

[2] J.E. Hopcroft and J.D. Ullman, Introduction to automata theory, languages and computation, Addison-Wesley, Reading, MA, 1979.

[3] O.H. Ibarra, A hierarchy theorem for polynomial-space recognition, *SIAM J. Comput.* **3** 3 (1974) 184-187.

[4] O.H. Ibarra and S.K. Sahni, Hierarchies of Turing machines with restricted tape alphabet size, *JCSS* **11** (1975) 56-67.

[5] J.I. Seiferas, Technique for separating space complexity classes, *JCSS* **14** (1977) 73-99.

[6] J.I. Seiferas, Technique for separating space complexity classes, *JCSS* **14** (1977) 100-129.

[7] M. Sipser, Halting space-bounded computations, *TCS* **10** (1980) 335-338.