1H－7

テープ記号数を制限した決定性TMと交代性TMの領域計算量について

岩本 宙造　岩間 一雄
九州芸術工科大学　京都大学工学部

1 はじめに

計算量クラスを分離することは，計算量理論の分野において最も重要な研究課題の一つである．決定性や非決定性，交代性の領域量や時間量の関係については，古くから数多くの研究結果が知られてきた．例えば，決定性時間量については，$t_2(n)$ が $t_1(n)\log t_1(n)$ よりも真に成長が速い関数とすると，DTIME($t_1(n)$) $\subsetneq$ DTIME($t_2(n)$) [3] なる階層性が存在することが知られている．また，決定性領域量や交代性時間量に対する決定性時間量の分離として，DTIME($t(n)$) $\subsetneq$ DSPACE($t(n)$) [4] や，DTIME($t(n)$) $\subsetneq$ ATIME($t(n)$) [1] などが示されている．一方，非決定性時間量に対する決定性時間量の分離では，DTIME(n) $\subsetneq$ NTIME(n) [6] が証明されているが，一般の $k > 1$ に対して DTIME(n^k) $\subsetneq$? NTIME(n^k) と拡張できるかについては未解決である．最近になって，交代性時間量に対する決定性時間量の分離は，DTIME($t(n)$) $\subsetneq$ $\Sigma_2(t(n))$ [2] のように交代数が2回に改善されている．その一方，テープ数が1の場合は，$(t(n))^{2/3}\log^2 t(n)$ 時間の交代性TMは $t(n)$ 時間の決定性TMを模倣できるという逆の結果も示されている [5]．

一方，決定性領域量については，非決定性や交代性といった機能を使わなくても，計算量をほんの少し増加させるだけで，クラスを分離できることが知られている．$s_2(n)$ が $s_1(n)$ より成長が真に速い関数とすると，DSPACE($s_1(n)$) $\subsetneq$ DSPACE($s_2(n)$) [3] なる階層性が存在することが知られている．テープ圧縮定理 [3] が成立するので，この階層性は最適であるといえる．本稿では，テープ記号数を固定して領域計算量クラスの厳密な分離を試みる．如何なる関数 $\psi(n) \neq O(1)$ に対しても，$s(n)$ 領域の決定性TMは $s(n)+\psi(n)$ 領域の交代性TMより能力が真に低いことを示す．

以下，節2にてモデルと結果を与え，節3にて証明の概略を与える．

Space Complexities of Deterministic and Alternating TMs with Restricted Tape Symbols
Chuzo IWAMOTO (Kyushu Institute of Design)
Kazuo IWAMA (Kyoto University)

2 モデルと結果

チューリング機械(TM)は，左端に特別の記号が書かれた右方向に無限の作業テープ，左右端に特別の記号が書かれた1本の入力テープをもつ．入力テープ記号と作業テープの記号は0,1のみである．（本稿の結果は，記号数 $m \geq 2$ の場合にも容易に拡張できる．）最初，作業テープの全てのマスには0が書かれている．TMが入力を受理するときは，作業ヘッドと入力ヘッドを左端に戻した状態で，唯一の受理状態 q_f で停止すると仮定する．長さ n の如何なる入力に対しても，作業テープ上に長さ $s(n)$ の文字列 $11\cdots1$ を出力するTMが存在するとき，$s(n)$ を領域強構成可能と呼ぶ．

定理1. $s(n)$ は $s(n) \leq 2^{(1-o(1))n}$ なる任意の領域強構成可能関数とし，$\psi(n)$ は，長さ $\psi(n)$ の文字列 $11\cdots1$ が $s(n)$ 領域決定性TMで生成できるという条件を満たす関数とする．このとき，如何に成長の遅い関数 $\psi(n) \neq O(1)$ に対しても，次の条件を満たす言語 L が存在する．(i) L を受理する $(s(n)+\psi(n))$ 領域の交代性TM T が存在する．(ii) 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 は停止するとは限らない．

3 証明の概略

$s(n)$ 領域の如何なる決定性TM T_x でも受理できない言語を受理できる $(s(n)+\psi(n))$ 領域交代性TM T を構成する．如何なるTM T_x に対しても，幾らでも長い符号化 x があるようにしたいので，符号化の後ろに十分長い文字列を挿入したものも T_x の符号化と見なす．ただし，挿入列（の一部）には $s(n)$ の値を2進数で表現した文字列が入っている．$s(n)$ は $2^{(1-o(1))n}$ で押えられるので，十分長い文字列を仮定すれば $s(n)$ の値の2進数表現を挿入列に含めることができる．

入力列 x が，上記の条件を満たしているかを判定するために，まず T は長さ $s(n)$ の文字列 $11\cdots1$ を生成する．次に，この文字列の長さを2進カウンタで以下のように数える．まず，$s(n)+1$ 番目のマスに値0を表わす定数長さの文字列（カウンタと呼ぶ）を生成する．カウンタの両端の境界を表わすため，カウンタの両端には長さが定数 c の文字列 $11\cdots1$ が置いておき，カウンタ中に0が

c個連続しないように，$c-1$ビットごとに0を挿入しておく．カウンタを左へ一つずつ動かしながら，カウンタ値を1ずつ上げていく．カウンタの桁が足りなくなるたびに，カウンタに$c-1$桁挿入する．カウンタが左端に着いたときのカウンタの内容は，$s(n)$の値を2進数で表現したものになっている．これを入力列中の挿入列と比較する．もし，入力列xが上記条件を満たしていなければ，Tは非受理状態で停止する．

TM Tは，T_xの初期計算状況が$s(n)$領域の計算で受理計算状況に行き着くかを以下のように判定する．T_xの計算状況のうち，$s(n)$領域という条件を満足する計算状況を頂点とし，T_xの遷移関数で遷移できる関係を有向枝とした有限の有向グラフを考える．このとき各受理計算状況は，必ず木の根になる．T_xの初期計算状況が，いずれかの受理計算状況を根とする木に属しているときに限り，T_xは入力を受理することになる．

(a) Tは∀分岐を利用して，全ての受理計算状況を以下のように生成する．Tは値$s(n)$をもつカウンタを作業テープの先頭に生成する．Tは∀分岐を利用して，作業テープの最初のマスの記号を推測して生成し，カウンタ値を1だけ減らして，カウンタの位置を右へ一つだけシフトする．この作業を繰り返して，$s(n)$個のマスのそれぞれの記号を∀状態で推測していく．（この作業ではカウンタの値の減少に伴ってカウンタの長さも短くなるので，値が0になったときは，カウンタの長さも定数になっている．したがって，Tは$s(n)$より本質的に多くの領域を必要としない．）

次に，Tは$(s(n)+1)$番目以降のマスに長さ$\psi(n)$の二つのブロックを生成する．これらのブロックはT_xの遷移関数と現在の状態を蓄えるために用いる．これらのブロックは，T_xの模倣中は作業ヘッドの下に置かれるように常に移動する．上と同じ理由で，これらブロックには定数ビットごとに0が挿入されている．TはT_xが停止したときのヘッド位置を∀分岐で推測し，その場所にヘッドを置いた状況をT_xの最終状況と仮定して，それを根とする木に初期状況が含まれるかを調べる．これは，以下の手続きで実行される．(i) 上記二つのブロックを一マスだけ左にシフトし，∀分岐を用いて次の二つを実行する．(ii) TはT_xの今の作業ヘッド位置が最終計算状況であると仮定して，初期状況の探索を開始する．（探索方法は，次の段落で示す．）(iii) Tは(i)を実行する．ブロックが作業テープの左端に行き着いたら，Tは受理状態で停止する．

(b) いま，T_xの計算状況はc_1であるとする．Tはブロックに蓄えられた各遷移関数を左から順に見ていき，次の条件を満たす最初の遷移関数(δ_1とする)を見つける．その条件とは，δ_1に従ってc_2からc_1へと遷移するような計算状況c_2が存在することである．そして，∀分岐を用いて次の二つを実行する．(i) TはT_xの動作を逆方向に模倣し，計算状況をc_2に換える．(ii) Tは遷移関数をδ_1から右に見ていき，上の条件を満たす遷移関数を見つけて同じ手続きを繰り返す．もし，そのような遷移関数が見つからなければ，Tは受理状態で停止する．

以上の手続きを進めるときに，(1) ヘッド位置iが$s(n)$より小さい値に収まっているか，(2) 現在の計算状況は初期状況か否かを以下のようにチェックする．まず，Tは∃分岐を用いて次の(a)と(b)を実行する．(a) $i > s(n)$であると推測する．この推測が正しいか否かをチェックし，正しければ受理状態で，正しくなければ非受理状態で停止する．このチェックのために，T_xはi番目までのマスの数をカウンタを用いて数え上げて，入力テープ上の$s(n)$の値を表わす文字列と比較する．(b) $i \leq s(n)$であると推測する．Tは，∀分岐を用いて次の手続きと(2)を実行する．Tは$i \leq s(n)$という推測が正しいかをチェックし，正しければ受理状態で，正しくなければ非受理状態で停止する．(2) Tは，∀分岐を用いて次の(i)と(ii)を実行する．(i) Tは現在のT_xの状況が初期状況か否かをチェックし，そうならば非受理状態で停止し，そうでなければ受理状態で停止する．(ii) Tは前段落の説明した探索手続きを続ける．

補題1. Tが入力xを受理するのは，T_xの初期状況が，如何なる受理計算状況を根とする木にも含まれていないときに限られる（つまり，T_xがxを受理しないときに限られる）．

補題2. $(s(n)+\psi(n))$領域の交代性TM Tが受理する言語は，$s(n)$領域の如何なる決定性TM T_xでも受理できない．

参考文献

[1] P.W. Dymond et al., Speedups of deterministic machines by synchronous parallel machines, *Proc. IEEE FOCS* (1983) 336-343.

[2] S. Gupta, Alternating time versus deterministic time: a separation, *Proc. IEEE FOCS* (1993) 266-277.

[3] J. Hartmanis et al., Hierarchies of memory limited computations, *Proc. IEEE Symp. on Switching Circuit Theory and Logical Design* (1973) 179-190.

[4] J.E. Hopcroft et al., On time versus space, *J. Assoc. Comput.*, **24** (1977) 332-337.

[5] W. Maass et al., Speed-up of Turing machines with one work tape and a two-way input tape, *SIAM J. Comput.* **16** 1 (1987) 195-202.

[6] W.J. Paul et al., On determinism versus non-determinism and related problems, *Proc. IEEE FOCS* (1983) 429-438.