

非決定性同時計算量について

東京女子大学 守屋悦朗(Etsuro Moriya)

東 海 大 学 岩田茂樹(Shigeki Iwata)

電気通信大学 笠井琢美(Takumi Kasai)

1. はじめに

問題の複雑さの計算量については、時間量、領域量、テープ上でのヘッドの反転数など多くの測度について研究がなされているが、これらを同時に考えたときの計算量については現在のところさほど多くの研究はなされていない [5,7-11]。すなわち、従来の研究では、例えば反転数限定のチューリングマシンを、領域量限定のチューリングマシンでシミュレートする際には時間量については全く考慮を払わなかったし、また、時間量限定のチューリングマシンでシミュレートする際には領域量については全く考慮しないのが普通であって、時間量と領域量を同時に限定したチューリングマシンを考えるということにはなかった。従って、各種の計算量を同時に限定するという“同時計算量”についてはあまり良いトレードオフは知られていない。

本稿では、読み出し専用の1本の入力テープと1本の記憶テープとを持つオフライン型非決定性チューリングマシンに対して、その時間量、領域量、作業テープ上でのヘッドの反転数に関する同時計算量を考える。定理の系として、例えば次の結果が得られる。

$$(1) \quad \text{NTIME.NSPACE}(\text{poly}, \text{polylog}) = \text{NREVERSAL.NSPACE}(\text{polylog}, \text{poly})$$

$$(2) \quad \text{NP} = \text{NTIME.NSPACE}(\text{poly}, \text{poly}) = \text{NREVERSAL.NSPACE}(\text{poly}, \text{poly})$$

$$(3) \quad \text{PSPACE} = \text{NREVERSAL}(\text{poly})$$

$$(4) \quad \text{NSPACE}(\text{polylog}) = \text{NREVERSAL}(\text{polylog})$$

ただし、 $X.Y(p, q)$ は、資源 X を p 限定かつ同時に資源 Y を q 限定したチューリングマシンによって受理される言語のクラスを表わし、 $\text{poly} = \{ n^i : i \geq 0 \}$, $\text{polylog} = \{ (\log n)^i : i \geq 0 \}$ である。(2)は別の考察からも容易に示される。(3)(4)は既知の結果である[3]。

2. 準備

本稿で考える計算のモデルは、1本の読み出し専用2方向入力テープと1本の記憶テープとを持つ非決定性のチューリングマシン(以下 NTM と略す)である。また、本稿で考える関数 $R(n), S(n), T(n)$ などは自然数から実数への単調増加関数である。

M を NTM とし、 $R(n)$ を関数とする。入力 x に対して、記憶テープ上でヘッドが高

だか $R(|x|)$ 回しか反転しないような計算過程を、 x に対する $R(n)$ 反転限定の計算過程という。任意の $x \in L(M)$ に対して、 $R(n)$ 反転限定の受理計算過程が存在するとき、 M は $R(n)$ 反転限定であるという。入力テープ上の反転数はカウントしないことに注意したい。

$$\text{NREVERSAL}(R(n)) = \{ L(M) : M \text{ は } O(R(n)) \text{ 反転限定の NTM } \}$$

と定義する。 $S(n)$ 領域限定性、 $T(n)$ 時間限定性、 $\text{NSPACE}(S(n))$ も同様に通常通り定義される。Greibach[3] は $\text{NREVERSAL}(R(n)) = \text{NSPACE}(R(n) \log n)$ を証明した。

ここでは次のような同時計算量を考える。入力 x に対して、 $x \in L(M)$ ならば $R(n)$ 反転限定かつ $S(n)$ 領域限定の M による受理計算過程が存在するとき、 M は $R(n)$ 反転 $S(n)$ 領域同時限定であるといい、そのようなチューリングマシンの受理する言語のクラスを次のように定義する：

$$\text{NRESP}(R(n), S(n)) = \{ L(M) : M \text{ は } O(R(n)) \text{ 反転 } O(S(n)) \text{ 領域同時限定の NTM } \}$$

同様に、 $T(n)$ 時間 $S(n)$ 領域同時限定の言語のクラスを

$$\text{NTISP}(T(n), S(n)) = \{ L(M) : M \text{ は } O(T(n)) \text{ 時間 } O(S(n)) \text{ 領域同時限定の NTM } \}$$

と定義する。

山本・野口[11] は、 $\text{NRESP}(R(n), S(n)) = \text{NRESP}(S(n), R(n))$ を証明した。Hong[5] は、 $R(n) \cdot S(n) = o(n)$ ならば $\text{NRESP}(R(n), S(n)) = \text{NRESP}(0, 1)$ 、すなわち、 $\text{NRESP}(R(n), S(n))$ は正則集合のクラスと一致することを証明した。

1次元非決定性セルオートマトン（以下、単にセルオートマトンと呼ぶ）とは $A = (Q, \delta, x_1 x_2 \dots x_n, F)$ のことをいう。ここに、 Q は状態の有限集合（ Q は、休止状態と呼ばれる特殊な元 B を含んでいるとする）、 $\delta: Q^3 \rightarrow 2^Q$ は遷移関数、 $x_1 x_2 \dots x_n$ ($x_i \in Q - \{B\}$) は初期様相を表わし、 $F \subseteq Q$ は最終状態の集合である。

A の様相とは関数 $c: \mathbb{Z} \rightarrow Q$ のことである。とくに、 $c(i) = x_i$ ($1 \leq i \leq n$) 以外の点では $c(i) = B$ であるような様相を $c = x_1 x_2 \dots x_n$ と略記する。すべての i に対して $c(i) \in F$ である様相を最終様相と呼ぶ。 A は初期様相から計算を始め、次に定義されるように δ に従って計算を進め、最終様相に到達すると停止する。すなわち、様相の上の2項関係 $\vdash_A$ ($\vdash$ と略記する) を

$$c \vdash_A c' \iff \forall i \in \mathbb{Z} [c'(i) \in \delta(c(i-1), c(i), c(i+1))]$$

と定義する。任意の様相 c に対して $c \vdash^0 c$ 、また、非負整数 i と様相 c_1, c_2 に対して $c_1 \vdash^{i+1} c_2 \iff \exists c_3 [c_1 \vdash^i c_3 \vdash c_2]$ と定義する。初期様相 c_0 に対して $c_0 \vdash^k c$ となる最終様相 c が存在するとき、 A は k ステップ限定であるという。

通常のセルオートマトンでは $\delta(B, B, B) = \{B\}$ という制約があるが、ここではそのような制限はしない。その代りに、 $B \in \delta(@, B, @)$ であるとする。 $@$ は任意の状態を表わす（以下でも、 $@$ はこのような意味で用いる）。従って、 B 以外の状態が無限個現われる様相も存在しうるが、以下ではそのような様相は最終様相へ到達しえないようなセルオート

マトンを考える。

様相 c は、 $\{i : c(i) \neq B\} \subseteq \{j+1, j+2, \dots, j+k\}$ となる j, k が存在するとき k セル限定であるという。初期様相 c_0 を持つセルオートマトン A が k セル限定であるとは、

$$\exists c_1, \dots, \exists c_m [c_0 \vdash c_1 \vdash \dots \vdash c_m \quad \& \quad c_m \text{ は最終様相} \\ \& \quad \forall i (0 \leq i \leq m) [c_i \text{ は } k \text{ セル限定}]]$$

が成立することをいう。 k セル限定セルオートマトンにおいては、連続する k 個のセルだけを考えればよいことに注意したい。

3. NRESP と NTISP の関係

補助定理 1 M は $R(n)$ 反転 $S(n)$ 領域同時限定 NTM とする。このとき、入力 x に対して次を満たすセルオートマトン A_x (のコード) を出力する決定性 $\log n$ 領域限定変換機 M_1 が存在する。

(1) $x \in L(M) \iff A_x$ は $2R(|x|)$ ステップ限定

(2) A_x は $S(|x|)$ セル限定

証明 M の状態集合を Q 、入力アルファベットを Σ 、テープアルファベットを Γ とする。 $x = x_1 x_2 \dots x_n$ ($x_i \in \Sigma$) に対して A_x を次のように定義する。

A_x の状態集合は次の 5 種類の状態より成る。

(i) Γ の元 ($B \in \Gamma$ とする)

(ii) $\langle q, i, a \rangle$ 、ただし $q \in Q, 1 \leq i \leq n, a \in \Gamma$

(iii) a' 、ただし $a \in \Gamma$ (各 $a \in \Gamma$ に対して、 a' は新しい記号とする)

(iv) $\langle p, i, a, q, j, b, s \rangle$ 、ただし $p, q \in Q; 1 \leq i, j \leq n; a, b \in \Gamma;$

$s \in \{\text{left to right, right to left, left end, right end, left start, right start}\}$

(v) dead

(i) と (ii) の状態で M の様相を表わす。すなわち、

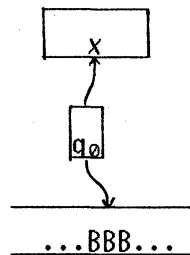
$$a_1 a_2 \dots a_{j-1} \langle q, i, a_j \rangle a_{j+1} \dots a_n \quad (*)$$

は、 M の現在の状態が q 、入力ヘッドの位置が i 、記憶テープの内容が $a_1 a_2 \dots a_n$ で、記憶テープのヘッドは a_j を読んでいることを意味する。 A_x は guess phase と check phase を交互にくりかえす。 $(*)$ の様相にあるとき、 A_x は guess phase にある。 A_x は check phase において、 M の 1 つのヘッド反転から次のヘッド反転までの間 (1 reversal) をシミュレートする。すなわち、 A_x の各セルは (iii)-(v) のどれかの状態をとる。

状態 (iii) は、このセルに対応する M の記憶テープの内容が a であることを意味し、状態 (iv) は、このセルに対応する M の記憶テープの内容が a であり、入力ヘッドの位置 i かつ状態 p のとき記憶テープのヘッドはこのセル (に対応する記憶テープ上のセル) に入り、引続く何 moves かをこのセルに停滞した後、最終的にはこのセルに b を書きこんで、入力ヘッド位置 j 状態 q でこのセルから出ていくことを意味する。 $s = \text{left}$

to right (right to left) は、このセルに左から入り右に出ていく (右から入り左に出ていく) ことを示す。left start は、このセルにおいてヘッドが左向きから右向きへ反転し、以後その右に位置するセルでは left to right 型のヘッド移動がなされること、すなわち、1 reversal の左開始点であることを示し、right end は、この reversal の終点、すなわち、ヘッドが右向きから左向きへ反転するセルであることを示す。right start, left end についても同様である。この check phase において、 A_x は guess した状態が M の動作に矛盾しない正当なものであるかどうかを両隣のセルを見てチェックし、もし不当なものであったら dead 状態に入り、以後ずっと dead 状態を続ける。

A_x の初期様相は $\langle q_0, 1, B \rangle$ である。 q_0 は M の初期状態。すなわち、 A_x は M の初期様相



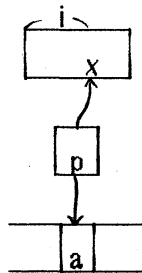
からシミュレーションを始める。

Guess phase での A_x の状態遷移関数は次のように定義される。

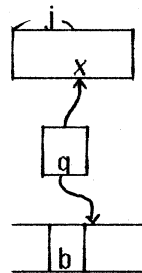
(I) 各 $a \in \Gamma$ に対して、 $\delta(@, a, @)$ は次の状態を含む：

(i) a'

(ii) M において



から



へ遷移しうる (以後、これ

$(1 \leq i, j \leq n; p, q \in Q; a, b \in \Gamma)$

を $(i, pa) \vdash (j, bq)$ と書くことにする。 $(i, pa) \vdash (j, qcb)$ etc. も同様に定義される。) ならば、 $\langle p, i, a, q, j, b, \text{left to right} \rangle$ と $\langle p, i, a, q, j, b, \text{left end} \rangle$ を含む。

(iii) $(i, pa) \vdash (j, qb)$ ならば、 $\langle p, i, a, q, j, b, \text{right to left} \rangle$ と $\langle p, i, a, q, j, b, \text{right end} \rangle$ を含む。

(II) 各 $\langle p, i, a \rangle$ に対し、 $\delta(@, \langle p, i, a \rangle, @)$ は次の状態を含む：

(i) $(i, pa) \vdash (j, bq)$ ならば $\langle p, i, a, q, j, b, \text{left start} \rangle$ を含む。

(ii) $(i, pa) \vdash (j, qb)$ ならば $\langle p, i, a, q, j, b, \text{right start} \rangle$ を含む。

A_x が guess phase $a_1 a_2 \dots a_{j-1} \langle p, i, a_j \rangle a_{j+1} \dots a_s$ にあるとき、すなわち M が

り永久にループする。Guess が成功だったときは、シミュレートした 1 reversal の終点セルが次の 1 reversal の開始セルとなる ((VI) による)。最後に、M が x を受理したら A_x は最終様相に入って停止する。すなわち、 A_x の最終状態の集合 $= \{B\} \cup \{ \langle q, i, a \rangle : q \in F, 1 \leq i \leq n, a \in \Gamma \}$ と定義する。F は M の最終状態の集合である。

A_x は、入力 x に対する M の計算における 1 reversal を 2 ステップでシミュレートし、M は $R(n)$ 反転限定なので、「 $x \in L(M) \iff A_x$ は $2R(|x|)$ ステップ限定」である。また、M は $S(n)$ 領域限定なので A_x は $S(|x|)$ セル限定である。

入力 x に対して、 M_1 は次のように A_x (のコード) を出力する。 M_1 は (I)-(VII) に従って $\delta(p, q, r) = \{e_1, \dots, e_j\}$ を計算し、 $\{(p, q, r, e_1), \dots, (p, q, r, e_j)\}$ を出力していく。(I)(ii)(iii) および (II) 以外の場合については、 δ の計算は明らかに DSPACE(1) でできる。(I)(i) の場合、 i, j, p, q, a, b を固定したときには、 $(i, pa) \vdash (j, bq)$ となるか否かの判定は DSPACE(1) でできる。従って、それらを動かした場合 ($1 \leq i, j \leq n$) に必要な領域量は DSPACE(log n) で充分である。(I)(ii), (II) についても同様。よって、 M_1 は log n 領域限定である。(証明終)

補助定理 2 セルオートマトン A と整数 k を入力として次を満たすセルオートマトン A' を出力する、決定性 log n 領域限定変換機 M_2 が存在する。

(3) A が k ステップ限定かつ m セル限定

$\iff A'$ が $2m$ ステップ限定かつ k セル限定

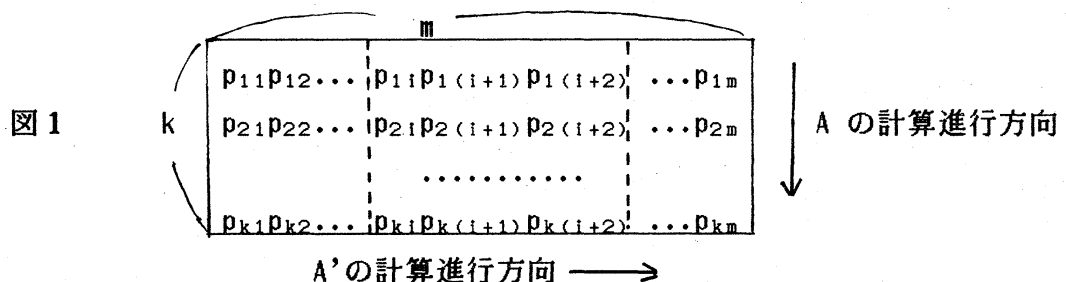
証明 $A = (Q, \delta, q_1 q_2 \dots q_s, F)$ とする。 A' の状態集合は次のように 5 種の状態より成る：

$$\begin{aligned} & \{(p_1, p_2, p_3) : p_1, p_2, p_3 \in Q\} \cup \{(p_1, p_2) : p_1, p_2 \in Q\} \\ & \cup \{(q_i, q_{i+1}, q_{i+2}, i) : 1 \leq i \leq s\} \cup \{(q_i, q_{i+1}, i) : 1 \leq i \leq s\} \\ & \cup \{r_0, r_1, r_2, \text{dead}\} \end{aligned}$$

ただし、 $i > s$ に対しては $q_i = B$ と考える。 A' の初期様相は $r_0 r_1^{-1} r_2$ とする。A における k ステップ限定かつ m セル限定の計算過程

$$C_1 \vdash C_2 \vdash \dots \vdash C_k$$

を考える。 $C_j = p_{j1} p_{j2} \dots p_{jm}$ ($p_{ji} \in Q$) とする。とくに、 $p_{ji} = q_i$ ($1 \leq i \leq s$), $p_{ji} = B$ ($i > s$) である。 A' は A の計算を図 1 のようにシミュレートする。



A' は guess phase と check phase を交互に繰り返す。 A' は、guess phase において

A の連続する3コマ (図の点線内) を guess し、check phase においてこれが A の動きに矛盾しないことをチェックする。すなわち、A' は $2i-1$ ステップ目で guess phase に入り、状態

$$(q_i, q_{i+1}, i)(p_{2i}, p_{2(i+1)})(p_{3i}, p_{3(i+1)}) \dots (p_{ki}, p_{k(i+1)})r_2$$

をとる。ここで A' は A の $i+2$ 列目

$p_{2(i+2)}$
$p_{3(i+2)}$
.....
$p_{k(i+2)}$

を guess し、check phase に入り、様相

$$(q_i, q_{i+1}, q_{i+2}, i)(p_{2i}, p_{2(i+1)}, p_{2(i+2)}) \dots (p_{ki}, p_{k(i+1)}, p_{k(i+2)})r_2$$

をとる。ここで A' は $2 \leq j \leq k$ について、 $\delta(@, (p_{ji}, p_{j(i+1)}, p_{j(i+2)}), @)$ が $p_{j(i+1)}(i+1)$ を含んでいるかどうかをテストし、含んでいたらこのセルの状態を $(p_{j(i+1)}, p_{j(i+2)})$ に変え、第1番目のセルの状態を $(q_{i+1}, q_{i+2}, i+1)$ に変え、guess phase に戻る。

A' の状態遷移関数 δ' は次のように定義すればよい。

(I) 初期化

$$\delta'(B, r_0, @) = \{(q_1, q_2, q_3, 1)\}$$

$$\delta'(@, r_1, r) = \begin{cases} \{(p_1, p_2, p_3) : p_1, p_2, p_3 \in Q\}, & r \neq r_2 \text{ のとき} \\ \{(p_1, p_2, p_3) : p_1, p_2 \in Q \text{ かつ } p_3 \in F\}, & r = r_2 \text{ のとき} \end{cases}$$

$$\delta'(r_1, r_2, B) = \{r_2\}$$

(II) Check phase

$$\delta'(B, (q_i, q_{i+1}, q_{i+2}, i), @) = \begin{cases} \{(q_{i+1}, q_{i+2}, i+1)\}, & 1 \leq i \leq s \text{ のとき} \\ \{(B, B, i)\}, & i = s+1 \text{ のとき} \end{cases}$$

$$\delta'((p_1, p_2, p_3), (p'_1, p'_2, p'_3), @) = \begin{cases} \{(p'_2, p'_3)\}, & p'_2 \in \delta(p_1, p_2, p_3) \text{ のとき} \\ \{\text{dead}\}, & \text{その他のとき} \end{cases}$$

$$\delta'((B, B, B), r_2, B) = \{r_2\}$$

(III) Guess phase

$$\delta'(B, (q_i, q_{i+1}, i), @) = \begin{cases} \{(q_i, q_{i+1}, q_{i+2}, i)\}, & 1 \leq i \leq s \text{ のとき} \\ \{(B, B, B, i)\}, & i = s+1 \text{ のとき} \end{cases}$$

$$\delta'(@, (p_1, p_2), r) = \begin{cases} \{(p_1, p_2, p_3) : p_1, p_2, p_3 \in Q\}, & r \neq r_2 \text{ のとき} \\ \{(p_1, p_2, p_3) : p_1, p_2 \in Q \text{ かつ } p_3 \in F\}, & r = r_2 \text{ のとき} \end{cases}$$

これ以外の組み合わせに対しては、 $\delta'(@, p, @) = \{p\}$ とする。

最後に、A' の最終状態の集合 $= \{(B, B, s+1), (B, B), r_2, B\}$ とする。

(3) が成立することは明らかである。また、A と k から A' を決定性 $\log n$ 領域で作れることは明らかであろう。(証明終)

補助定理3 セルオートマトン A (のコード) を入力として、 A が停止するか否かを判定する NTM M_3 で次を満たすものが存在する。

(4) A が k セル限定かつ m ステップ限定

$\iff$ ある多項式 p に対して

M_3 は $k \cdot \log n$ 領域 $m \cdot k \cdot p(n)$ 時間同時限定

証明 A の1つの状態は、(例えば2進コーディングすることにより) $\log n$ 領域で表現できることに注意する(これを“ $\log n$ 表現”ということにする)。 M_3 は2本の記憶テープを持ち、初め M_3 は入力 A から A の遷移関数部分の $\log n$ 表現を求めて第1テープに書きこむ。これらは作業用に非決定性 $\log n$ 領域を使えばでき、よって、入力の長さの多項式時間でできる。次に M_3 は、第1テープに書かれた A の遷移関数に従って、第2テープ上で A の動作のシミュレーションを行なう。 M_3 がこれに使う領域量は、 A が k セル限定なら $k \cdot \log n$ 領域で充分であり、 A の1ステップのシミュレーションをするのに M_3 は (A の非休止状態の数) $\times$ (A の遷移関数部の長さ) に比例する時間あれば充分なので、 A が m ステップ限定なら M_3 が使う時間は総計 $O(m \cdot k \cdot n \cdot \log n)$ あれば充分である。 $T(n)$ 時間限定の2テープ NTM は $T(n)^2$ 時間限定の1テープ NTM でシミュレートできるから、 M_3 を求める条件を満たす NTM とできる。(証明終)

定理1 $S(n)$ を任意の関数、 $R(n)$ をある多項式 q に対して $R(n) \leq q(n)$ となる全時間構成可能な関数とする。このとき、

$$NRESP(R(n), S(n)) \subseteq NTISP(R(n)S(n)t(n), R(n)\log n)$$

を満たす多項式 t が存在する。

証明 $L \in NRESP(R(n), S(n))$, $L \subseteq \Sigma^*$ とする。 $x \in \Sigma^*$ に対して、 $x \in L$ であるか否かは次のような非決定性の手続き M_4 で判定できる(図2参照)。

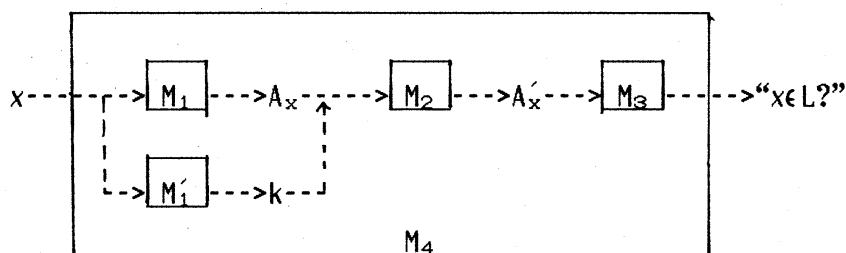


図2

$|x| = n$, $k = R(n)$, $m = S(n)$ とおく。先ず補助定理1により A_x を作ると

$x \in L \iff A_x$ は $2k$ ステップ限定かつ m セル限定

である。 M_1 は決定性 $\log n$ 領域限定であるから、ある多項式 p_1 に対して $|A_x| \leq p_1(n)$ かつ M_1 は $p_1(n)$ 時間限定である。ここに、 $|A_x|$ は A_x のコードの長さを表わす。 $k = R(n)$ は全時間構成可能であるから、任意の x に対して、 k を unary な表現で出力する k 時間 k 領域同時限定の変換機 M_1' が存在する。次に補助定理2により、 k, A_x から

A_x を作れば

$x \in L \iff A_x$ は $2m$ ステップ限定かつ $2k$ セル限定

である。 M_2 は決定性 $\log n$ 領域限定であるから、 $|A_x| \leq p_2(|A_x| + k)$ かつ M_2 が $p_2(|A_x| + k)$ 時間限定となる多項式 p_2 が存在する。 $|A_x| + k \leq p_1(n) + q(n) = p_3(n)$, $p_4(n) = p_2(p_3(n))$ とおくと、 $|A_x| \leq p_4(n)$ である。

最後に、 $x \in L$ であるかどうか（すなわち、 A_x が $2m$ ステップ限定かつ $2k$ セル限定であるかどうか）を補助定理 3 により判定する。これは非決定性 $2k \cdot \log |A_x| = O(R(n) \log n)$ 領域ででき、 $x \in L$ （すなわち、 A_x が $2m$ ステップ限定）なら M_3 は $2m \cdot k \cdot p(p_4(n)) = O(S(n)R(n)t(n))$ 時間以内に停止しうる。ここに $t(n) = p(p_4(n))$ は多項式である。すなわち、 M_4 は $R(n) \log n$ 領域 $R(n)S(n)t(n)$ 時間同時限定である。（証明終）

注意 定理 1 は、セルオートマトンを使わない直接的なシミュレーションで証明することもできる[7]。また、もっと一般に、任意の $R(n)$, $S(n)$ に対して

$$\text{NRESP}(R(n), S(n)) \subseteq \text{NTISP}(R(n)S(n)n^2 \log n, R(n) \log n)$$

となることが最近証明できた[8]。

定理 2 任意の $T(n)$, $S(n)$ に対して

$$\text{NTISP}(T(n), S(n)) \subseteq \text{NRESP}(S(n), T(n))$$

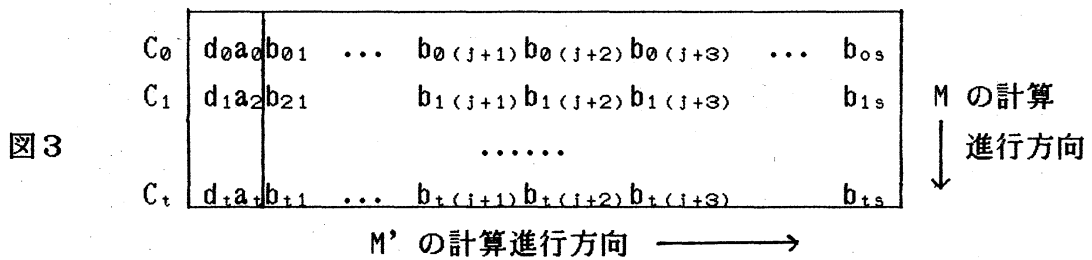
が成り立つ。

証明 $L \in \text{NTISP}(T(n), S(n))$, $L \subseteq \Sigma^*$ とし、 L を受理する $T(n)$ 時間 $S(n)$ 領域同時限定の NTM を M とする。 M のテープアルファベットを Γ とするとき、 M の様相を $\{-1, 0, 1\} \Sigma^* (Q \times \Gamma) \Gamma^*$ の元 $\text{dau}(q, b)v$ で表わす。これは M が現在状態 q にいて、入力テープ上で記号 a を、記憶テープ上で記号 b を読んでおり、記憶テープの内容は ubv であり、入力テープ上のヘッドが $d \in \{-1, 0, 1\}$ の方向に動いてこの様相になったことを表わす。初期様相においては $d=0$ であるとする。 $x = x_1 x_2 \dots x_n$ ($x_i \in \Sigma$) を M への入力とし、 $t \leq T(n)$, $s = S(n)$ とする。

$$C_0 \vdash C_1 \vdash \dots \vdash C_t, \quad C_i = d_i a_i b_{i1} b_{i2} \dots b_{is} \quad (0 \leq i \leq t)$$

を入力 x に対する M の計算過程とする。ただし、 b_{ij} のうちの 1 つは $Q \times \Gamma$ の元、他の b_{ij} は Γ の元である。また、 $d_0 = 0$, $b_{01} = (q_0, B)$, $b_{0j} = B$ ($2 \leq j \leq s$) である。ここに、 q_0 は M の初期状態、 B はブランク記号である。 $m(i) = \sum_{k=0}^i d_k$ とすると、 $m(i)$ は様相 C_i における入力ヘッドの位置を表わすことに注意したい。すなわち、 $a_i = x_{m(i)}$ である。

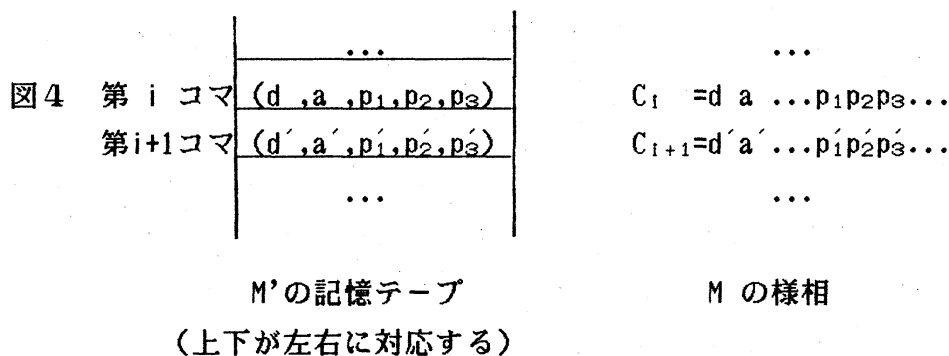
L を受理する $O(S(n))$ 反転 $T(n)$ 領域同時限定の NTM M' を構成する。 M' のテープ記号は、 $\{-1, 0, 1\} \times \Sigma \times \{(\Gamma^0 Q \times \Gamma)^{2^0} (\Gamma^0 Q \times \Gamma)^{2^1} \dots\}$ の元である (d, a, p_1, p_2) なる形の記号か または (d, a, p_1, p_2, p_3) なる形の記号である。 M' は M の計算を図 3 に示したようにシミュレートする。



初期化 初めに M' は t を guess し、記憶テープ上に記号列 $d' = \alpha_0 \alpha_1 \dots \alpha_t$ (α_i は $\{-1, 0, 1\} \times \Sigma \times (\Gamma \cup Q \times \Gamma)^3$ の元) を guess して書きこむ。 $\alpha_i = (d_i, a_i, p_{i1}, p_{i2}, p_{i3})$ は次のことを意味する。まず、 p_{i1}, p_{i2}, p_{i3} のうちの高だか1つだけが $Q \times \Gamma$ の元であり、他は Γ の元である。 $p_{i1} p_{i2} p_{i3}$ は、様相 C_i における M の記憶テープ上の連続する3コマの内容 $b_{i(j+1)} b_{i(j+2)} b_{i(j+3)}$ に等しく、もし $p_{ik} = (q, b)$ ならば C_i において記憶テープヘッドは $b = b_{i(j+k)}$ を読んでおり、そのときの状態は q である。また、 d は C_i におけるヘッドの移動方向を、 a は C_i における入力テープ上で読まれている記号を意味する。よって、 $\alpha_0 = (0, x_1, (q_0, B, B, B))$ でなければならない。 M' は、 α_i の第2成分 a_i が M の動きと矛盾しないこと(入力 x に対する整合性)を $\alpha_0, \alpha_1, \dots, \alpha_t$ をこの順に調べることによってチェックする。すなわち、 α_i の第1成分 $d_i \in \{-1, 0, 1\}$ に従って M' の入力ヘッドを移動し、そのときの入力記号と a_i が一致するかどうか調べればよい。 α がこのチェックにパスしたら、記憶テープのヘッドを左端に戻す。このチェックは1回のヘッド反転で行なえる(入力ヘッドの反転数はカウントしないことに注意)。

シミュレーション 初期化後、 M' は check phase と guess phase を交互にくり返す。第 $2j$ ステップは guess phase、第 $2j+1$ ステップは check phase である。

(Check phase) M' は記憶テープを左から右へ読みながら、隣り同士のコマの内容が M の動きと両立するかどうかチェックする。いま、第 i コマの内容が (d, a, p_1, p_2, p_3) 、第 $i+1$ コマの内容が $(d', a', p'_1, p'_2, p'_3)$ であったとする(図4参照)。



M' の記憶テープの2つの隣り合うコマ(図4左)が M の動き(図4右)と両立すると

は、次のいずれかが成立することをいう。

(i) p_1, p_2, p_3 および p'_1, p'_2, p'_3 のいずれも Γ の元である。(M の記憶ヘッドはこの範囲内にないことを意味する。)

(ii) $d' = -1, p_3 = (q, b), \delta(q, a, b) \ni (q', b', -1), p'_3 = b', p'_2 = (q', p_2), p'_1 = p_1$ である。

ここに、 δ は M の遷移関数である。

(iii) $d' = 0, p_2 = (q, b), \delta(q, a, b) \ni (q', b', 0), p'_2 = (q', b'), p'_1 = p_1, p'_3 = p_3$

(iv) $d' = 1, p_1 = (q, b), \delta(q, a, b) \ni (q', b', 1), p'_2 = (q', p_2), p'_1 = b', p'_3 = p_3$

もし (d, a, p_1, p_2, p_3) と $(d', a', p'_1, p'_2, p'_3)$ が両立したなら、 (d, a, p_1, p_2, p_3) を (d, a, p_2, p_3) に書きかえ、そうでないなら拒否して停止する。隣りのコマ同士の両立性のチェックは有限状態部を使うだけでチェックできる。記憶テープ上の $t+1$ 個のコマのすべてがこのチェックに合格したら、ヘッドを記憶テープの左端に戻し、guess phase に入る。よって check phase でのヘッド反転は2回である。このチェックの最終段階 $i+1=t$ において $p_2 \in F$ (F は M の停止状態の集合)であったら、このこと(停止状態通過ということにする)を有限状態部に覚えておく。

(Guess phase) この phase の始まり時には、記憶テープの各セルに書かれているのは (d, a, p_1, p_2) の形の記号である。M' は記憶テープ上を左から右にヘッドを動かしながら、 $\Gamma^4 Q \times \Gamma$ の元 p_3 を guess し、 (d, a, p_1, p_2) を (d, a, p_1, p_2, p_3) に代えていく。ただし、左端のコマに対しては $p_3 = B$ とする。これによって M の記憶テープとの整合性を1コマ分進めたことになる。Guess phase においては、2回のヘッド反転を行なう。

後処理 Check phase において、すべてのコマについてそこに書かれていた記号 (d, a, p_1, p_2, p_3) が $p_1 = p_2 = p_3 = B$ であったとき M' は停止し、この check phase がパスしかつ“停止状態通過”であったなら M' は入力を受理し、そうでなかったら拒否する。

上記の構成により、 $L(M') = L$ であり、かつ M' は $2S(n)+1$ 反転 $T(n)$ 領域同時限定である。(証明終)

系 $NTISP(poly, polylog) = NRESP(polylog, poly)$.

参考文献

- [1] P.C.Fischer, The reduction of tape reversals for off-line one-tape Turing machines, JCSS 2 (1968), 136-147.
- [2] P.C.Fischer, J.Hartmanis and M.Blum, Tape reversal complexity hierarchies, IEEE Conf. Record of 9th SWAT (1968), 373-382.
- [3] S.A.Greibach, Visits, crosses, and reversals for nondeterministic off-line machines, Inform. Contr. 36 (1978), 174-216.
- [4] J.Hartmanis, Tape-reversal bounded Turing machine computations, JCSS 2 (1968), 117-135.

- [5] J-W.Hong, A tradeoff theorem for space and reversal, Theor. Comput. Sci. 32 (1984), 221-224.
- [6] T.Kameda and R.Vollman, Note on tape reversal complexity of languages, Inform. Contr. 17 (1970), 203-215.
- [7] 守屋、岩田、笠井、非決定性チューリングマシンの時間量、領域量、反転数の同時計算量について、信学技報 AL85-2, 1985. (予定)
- [8] E.Moriya, S.Iwata and T.Kasai, Tradeoffs between simultaneous complexity measures, in preparation.
- [9] N.Pippenger and M.J.Fischer, Relations among complexity measures, JACM 26 (1979), 361-381.
- [10] 関口、低いレベルの同時計算量について、信学技報 AL84-8, 1984.
- [11] 山本、野口、リバーサル計算上での ATM と NTM の能力の比較、信学技報 AL84-12, 1984.