

Title	空間限定 1 ペブル 2 次元チューリング機械の受理能力に関する研究
Author(s)	井上, 敦之
Citation	
Issue Date	2003-03
Type	Thesis or Dissertation
Text version	author
URL	<a href="http://hdl.handle.net/10119/1651">http://hdl.handle.net/10119/1651</a>
Rights	
Description	Supervisor:平石 邦彦, 情報科学研究科, 修士

修 士 論 文

空間限定 1 ペブル 2 次元チューリング機械の  
受理能力に関する研究

北陸先端科学技術大学院大学  
情報科学研究科情報システム学専攻

井上 敦之

2003 年 3 月

# 修士論文

## 空間限定 1 ペブル 2 次元チューリング機械の 受理能力に関する研究

指導教官 平石邦彦 助教授

審査委員主査 平石邦彦 助教授  
審査委員 浅野哲夫 教授  
審査委員 金子峰雄 教授

北陸先端科学技術大学院大学  
情報科学研究科情報システム学専攻

910011 井上 敦之

提出年月: 2003 年 2 月

## 目次

1. はじめに
2. 定義と記法
3. 決定性と非決定性の関係
4. 非決定性と交代性の関係
5. まとめ

謝辞

参考文献

付録

# 第1章 はじめに

計算機科学における主要な分野に計算量理論がある。計算量理論の主な目的は、以下の2点である。

- (1) 計算を高速化する方法, 或いは計算をより少ない記憶容量で行う方法をソフトウェア的 またはハードウェア的に考え出すこと。
- (2) 問題の本質的難しさを明らかにしていくこと。

目的(1)については、たとえば現在のコンピュータの基本的な計算モデルであるチューリング機械によってどのような問題が効率的に(すなわち、時間で言えば入力サイズの多項式時間で、記憶容量で言えば、例えば入力サイズの対数空間で)解けるか、あるいは効率的に解けないかを明らかにしたり、チューリング機械では効率的に解けない問題を効率的に解くための計算モデルとしてどのようなものが考えられるかなどを明らかにする研究が行われている。

目的(2)は、ある問題を解くのに、どの計算モデル上では、最低どれだけの記憶容量やステップ数が必要であるという形の命題を証明することである。このようなことがわかれば、計算モデルの能力も明確になり、ある問題を解かせるのに、どの計算モデルが必要であり、どの計算モデルは役立たないかの判断に利用できる。計算量理論では、このような問題の本質的むつかしさを、計算量クラスの包含関係を明らかにすることにより解明しようとするものである。

チューリング機械による判定問題は、チューリング機械による言語認識問題に置き換えて考えることが出来る。計算量としては、時間計算量と空間計算量が考えられる。チューリング機械の空間量に基づく計算量クラスとしては、用いられる空間量(記憶テープのコマ数)の違いと動作の決定性、非決定性、交代性の違いにより、種々の言語クラスが考えられているが、本研究では特に、同一の空間量を用いるときに、動作の決定性、非決定性、交代性の違いにより、チューリング機械の言語受理能力がどのように異なってくるかに着目する。

空間限定の決定性、非決定性、ならびに交代性1次元チューリング機械で受理される言語のクラスに関しては、

- (i)  $o(\log \log n)$  空間限定の場合、決定性、非決定性ならびに交代性はすべて同一の言語受理能力を有し、通常の有限オートマトンと同等である、

- (ii)  $\log\log n$  以上  $o(\log n)$  空間限定の場合、交代性は、決定性、非決定性より強い言語受理能力を有する、

ことが知られているが [?],  $\log\log n$  以上の空間限定の場合、非決定性は決定性よりも言語受理能力が強いかどうか、また  $\log n$  以上の空間限定の場合、交代性は非決定性よりも言語受理能力が強いかどうかは、分かっていない。

1次元チューリング機械の入力テープ上に1つのペブルの使用を許した1ペブル1次元チューリング機械においても、 $o(\log\log n)$  空間限定の場合、決定性と非決定性は同一の言語受理能力を有し、通常の有限オートマトンと同等であることが知られているが [?],  $\log\log n$  以上の空間限定の場合、非決定性は決定性よりも、また交代性は非決定性よりも言語受理能力が強いかどうかは分かっていない。

Blum と Hewitt は、2次元パターンの認識装置として2次元有限オートマトンを提案し、非決定性2次元有限オートマトンは決定性2次元有限オートマトンよりも強い受理能力を有することを示した [?]. 更に、井上、高浪、谷口は、2次元有限オートマトンに記憶テープを付加した2次元チューリング機械の性質を調べ、正方形テープに限定した場合でも、 $o(\log n)$  空間限定の2次元チューリング機械では、交代性は非決定性よりも、また非決定性は決定性よりも強い受理能力を有することを示した [?].

Blum と Hewitt は、2次元有限オートマトンの入力テープ上に1つのペブルの使用を許した1ペブル2次元有限オートマトンは、2次元有限オートマトンよりも強い受理能力を有することを示しているが [?], 1ペブル2次元有限オートマトンでは、決定性、非決定性、交代性の間にどのような受理能力の関係があるかは明らかにしていない。

本研究では、2次元チューリング機械の入力テープ上に1個のペブルの使用を許した1ペブル2次元チューリング機械 (p2-tm) を導入し、この計算モデルでどのように空間量限定されると、決定性、非決定性、交代性の間に受理能力の差が生じるかを調べる。中心の位置の記号が"1"であるような  $\{0, 1\}$  上の正方形テープの集合は、1ペブル2次元決定性有限オートマトンでは受理できるが [1],  $o(\log n)$  空間限定2次元決定性チューリング機械では受理できないことが知られている [?]. このことより、 $o(\log n)$  空間限定の場合、1ペブル2次元決定性チューリング機械は、2次元決定性チューリング機械より受理能力が高いことが分かる。

本論文の第2章では、本論文に関係のある諸定義と諸記法を与える。第3章では、空間限定1ペブル2次元チューリング機械における決定性と非決定性の受理能力の相違について考案し、正方形テープに限定された場合でも、 $o(\log n)$  空間限定の場合は、非決定性は決定性よりも受理能力が強いことを示す。第4章では、空間限定1ペブル2次元チューリング機械における非決定性と交代性の受理能力の相違について考案し、 $L(m, n) = f(m) + g(n)$  で、 $f(m) = o(\log m)$  かつ  $g(n) = o(\log n)$  のとき、 $L(m, n)$  空間限定1ペブル2次元チューリング機械では、交代性は非決定性より受理能力が強いことを示す ( $L(m, n)$  空間限定に関する定義については、第2章を参照されたい)。第5章では、まとめと今後の課題を与える。

## 第2章 定義と記法

### 2.1 2次元テーブル

$\Sigma$  を記号の有限集合とする。 $\Sigma$  上の2次元テーブルとは、 $\Sigma$  の要素からなる2次元矩形配列である。 $\Sigma$  上のすべての2次元テーブルから成る集合を  $\Sigma^{(2)}$  と記す。2次元テーブル  $x \in \Sigma^{(2)}$  に対し、 $x$  の行の数を  $l_1(x)$  と記し、 $x$  の列の数を  $l_2(x)$  と記す。また、正の整数  $m, n \geq 1$  に対し、 $\Sigma^{(m,n)} = \{x \in \Sigma^{(2)} \mid l_1(x) = m \wedge l_2(x) = n\}$  とする。各  $i, j (1 \leq i \leq l_1(x), 1 \leq j \leq l_2(x))$  に対し、 $x$  の  $i$  行  $j$  列の記号を  $x(i, j)$  と記す。また、 $1 \leq i \leq i' \leq l_1(x)$  かつ  $1 \leq j \leq j' \leq l_2(x)$  であるときのみ、 $x[(i, j), (i', j')]$  を、次の (i), (ii) を満たすような2次元テーブル  $z$  として定義する：

- (i)  $l_1(z) = i' - i + 1$  かつ  $l_2(z) = j' - j + 1$ ;
- (ii) 各  $k, r (1 \leq k \leq l_1(z), 1 \leq r \leq l_2(z))$  に対し、

$$z(k, r) = x(k + i - 1, r + j - 1).$$

行数が同一であるような二つの2次元テーブル  $x, y$  に対し、 $x$  の右に  $y$  を付加して得られる2次元テーブルを  $xy$  と記す。

### 2.2 1ペブル2次元チューリング機械

1ペブル2次元交代性チューリング機械 ( $p2-atm$ ) は、2次元交代性チューリング機械 [?] の入力テープ上に1個のペブルの使用を許した機械である。図1に示すように、 $p2-atm$   $M$  は、境界記号#で囲まれた読み取り専用矩形入力テープと、初期には空白の片側無限記憶テープを一つもつ。もちろん、 $M$  は制御部、入力ヘッド、ならびに記憶テープヘッドをもつ。 $M$  は動作の途中で入力テープ上に1個のペブルを置くことができる。直感的には、ペブルは、入力ヘッドが現在読んでいるコマ上に将来戻りたいときに、そのコマ上に”目印”をつけるために用いられる。入力ヘッドは、それが現在読んでいるコマ上にペブルが存在するか否かを検知することができ、そのコマからペブルを拾い上げたり、そのコマ上にペブルを置いたりすることができる。現在、入力ヘッドが読んでいるコマ以外のコマ上にペブルが置かれており、ペブルを拾い上げたいときには、 $M$  はペブルが置かれているコマ上に入力ヘッドを移動しなければならない。

$p2-atm$   $M$  は、形式的には次のような7項組  $M = (Q, q_0, U, F, \Sigma, \Gamma, \delta)$  で定義される。ここで、(1)  $Q = Q' \times \{0, 1\}$  は、制御部の状態の有限集合で、 $Q'$  は'内部状態の集合 ( $Q' \times \{1\}$ )

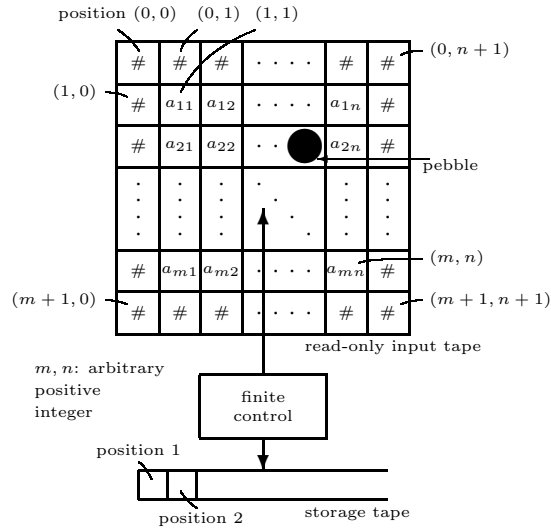


図 2.1: 1 ペブル 2 次元交代性チューリング機械.

に含まれる状態は、 $M$  が制御部にペブルを持っているときの状態に相当し、 $Q' \times \{0\}$  に含まれる状態は、 $M$  が入力テープ上のどこかのコマ上にペブルを置き、制御部にはペブルを持っていないときの状態に相当する。 (2)  $q_0 \in Q' \times \{1\}$  は、初期状態、 (3)  $U \subseteq Q$  は、全称状態の集合、 (4)  $F \subseteq Q$  は、受理状態の集合、 (5)  $\Sigma$  は、有限な入力アルファベット、 (6)  $\Gamma$  は、有限な記憶テープアルファベット ( $B \in \Gamma$  は、空白記号)、 (7)  $\delta : Q \times (\Sigma \cup \{\#\}) \times \{0, 1\} \times \Gamma \rightarrow P(Q \times (\Gamma - \{B\}) \times \{left, right, up, down, no\ move\} \times \{left, right, no\ move\})$  は、遷移関数。ここに、 $\# \notin \Sigma$  は境界記号であり、 $P(A)$  は集合  $A$  のすべての部分集合からなる集合を表す。

$Q - U$  に含まれる状態  $q$  は存在状態とよばれる。図 1 に示すように、読み取り専用入力テープと記憶テープの各コマには各々一つの位置が割り当てられる。遷移関数  $\delta$  は、 $M$  の現在の制御部の状態  $q$ 、入力ヘッドが読んでいるコマ上の記号  $\sigma$ 、入力ヘッドが読んでいるコマ上にペブルが置かれているかどうかの情報  $c$  ( $c = 1$  は、そのコマ上にペブルが置かれていることを表し、 $c = 0$  は、そのコマ上にペブルが置かれていないことを表す)、更には記憶テープヘッドが読んでいるコマ上の記号  $\gamma$  とによって、 $M$  の 1 ステップの動作を決める関数である。すなわち、もし

$$(q', \gamma', m_1, m_2) \in \delta(q, \sigma, c, \gamma)$$

であれば、 $M$  は 1 ステップの動作で、 (1) 制御部の状態を  $q'$  に変え、 (2) 記憶テープヘッドが読んでいるコマ上に記号  $\gamma'$  を印刷し、 (3) 入力ヘッドを 1 コマ  $m_1$  の方向に動かし ( $m_1 = left, right, up, down$  であれば、それぞれ左、右、上、下に動かし、また  $m_1 = no\ move$  であれば、動かさない)、更に (4) 記憶テープヘッドを 1 コマ  $m_2$  の方向に動か



す ( $m_2 = left, right$  であれば、それぞれ左、右に動かし、 $m_2 = no\ move$  であれば、動かさない) ことが可能である。この場合、

- $q \in Q' \times \{1\}$  かつ  $c = 0$  かつ  $q' \in Q' \times \{0\}$  は、制御部が持っているペブルを入力ヘッドが読んでいるコマ上に置いて状態  $q'$  に入ることを意味し、
- $q \in Q' \times \{1\}$  かつ  $c = 0$  かつ  $q' \in Q' \times \{1\}$  は、制御部が持っているペブルを入力ヘッドが読んでいるコマ上に置くことなしに状態  $q'$  に入ることを意味し、
- $q \in Q' \times \{0\}$  かつ  $c = 1$  かつ  $q' \in Q' \times \{1\}$  は、入力ヘッドが読んでいるコマ上にペブルがあり、そのペブルを拾い上げて状態  $q'$  に入ることを意味し、
- $q \in Q' \times \{0\}$  かつ  $c = 1$  かつ  $q' \in Q' \times \{0\}$  は、入力ヘッドが読んでいるコマ上にペブルがあり、そのペブルを拾い上げないで状態  $q'$  に入ることを意味し、
- $q \in Q' \times \{0\}$  かつ  $c = 0$  かつ  $q' \in Q' \times \{0\}$  は、入力ヘッドが読んでいるコマ以外のコマ上にペブルが置かれており、入力ヘッドが読んでいるコマ上にはペブルを置かないで状態  $q'$  に入ることを意味する。

$q \in Q' \times \{1\}$  かつ  $c = 1$  の組み合わせは考えられないので、このような組み合わせに対しては、任意の  $\sigma, \gamma$  に対し、 $\delta(q, \sigma, c, \gamma) = \phi$  ( $\phi$  は空集合を表す) とする。また、同様に、 $q \in Q' \times \{0\}$  かつ  $c = 0$  かつ  $q' \in Q' \times \{1\}$  であるような  $q, c, q'$  に対しては、任意の  $\sigma, \gamma, \gamma', m_1, m_2$  に対し、 $(q', \gamma', m_1, m_2) \notin \delta(q, \sigma, c, \gamma)$  とする。

境界記号 # を越えて入力テープからはみ出した場合、あるいは記憶テープヘッドが記憶テープから (左に動いて) はみ出した場合には、それ以後、機械  $M$  は停止するものとする。

2次元入力テープ  $x$  上での  $M$  の計算状況とは、対  $c = ((i, j), pebble - position, (q, \alpha, k))$  のことである。ここで、 $c$  の第1成分  $(i, j)$  は、入力ヘッドが読んでいるコマの位置を表し、 $c$  の第2成分  $pebble - position$  はペブルの置かれている入力テープ上のコマの位置を表し (制御部にペブルをもち、入力テープ上にペブルが置かれていないときには、 $pebble - position = no$  とする)  $(q, \alpha, k)$  は、それぞれ制御部の状態、記憶テープの非空白部分の内容、および記憶テープヘッドが読んでいるコマの位置を表す。計算状況  $c$  にかかわる状態  $q$  が全称 (存在、受理) 状態ならば、 $c$  は全称 (存在、受理) 計算状況であると言う。

2次元入力テープ  $x$  上での  $M$  の初期計算状況は、 $I_M = ((1, 1), no, (q_0, \lambda, 1))$  である。ここで、 $q_0$  は  $M$  の初期状態で、 $\lambda$  は空系列である。もし計算状況  $c'$  が入力テープ  $x$  上で  $M$  の1ステップにより計算状況  $c$  から導かれ得るならば、 $c'$  は  $x$  上での  $M$  の後継者と言い、 $c \mapsto_{M, x} c'$  と記す。 $c$  の  $x$  上での  $M$  の後継者の集合を  $Succ_{M, x}(c)$  で表す。 $Succ_{M, x}(c)$  が空集合であるような計算状況  $c$  を  $x$  上での  $M$  の停止計算状況とよぶ。以下、本論文では、受理計算状況は停止計算状況であるとする。

2次元入力テープ  $x$  上での  $M$  の計算木とは、その各節点が計算状況でラベルづけられた次のような木である：

- (1) 根のラベルは、初期計算状況  $I_M$  である。
- (2) 内部節点  $v$  のラベルが全称計算状況  $d$  であり、かつ  $Succ_{M, x}(d) = \{d_1, d_2, \dots, d_k\}$  な

らば、 $v$  は  $d_i = \text{label}(v_i)$  なるちょうど  $k$  個の子供  $v_1, v_2, \dots, v_k$  をもつ。ここで、 $\text{label}(v_i)$  は節点  $v_i$  のラベルを表す。

(3) 内部節点  $v$  のラベルが存在計算状況  $d$  ならば、 $v$  は  $\text{label}(v') \in \text{Succ}_{M,x}(d)$  なるちょうど1つの子供  $v'$  をもつ。

2次元入力テープ  $x$  上での  $M$  の受理計算木とは、すべての葉が受理計算状況によってラベルづけされているような  $x$  上での  $M$  の有限な計算木である。 $x$  上での  $M$  の受理計算木が存在するならば、 $M$  は  $x$  を受理すると言う。 $M$  によって受理されるようなすべての2次元テープの集合を  $T(M)$  と記す。

1ペブル2次元非決定性チューリング機械 ( $p2\text{-ntm}$ ) は、存在状態だけを持つような  $p2\text{-atm}$  であり、1ペブル2次元決定性チューリング機械 ( $p2\text{-dtm}$ ) は、その計算状況がすべて高々1つの後継者をもつような  $p2\text{-atm}$  である。

入力テープが正方形に限定されているような  $p2\text{-atm}(p2\text{-ntm}, p2\text{-dtm})$   $M$  を考える。いま、 $L(n) : N \rightarrow N$  ( $N$  は正の整数全体の集合) を1変数  $n$  の関数とする。 $n$  行  $n$  列 ( $n \geq 1$ ) のいかなる正方形の入力テープが与えられても、記憶テープの高々  $L(n)$  個のコマしか用いないような場合、 $M$  は  $L(n)$  空間限定であると言われる。以下、入力テープが正方形に限定されているような  $p2\text{-atm}(p2\text{-ntm}, p2\text{-dtm})$  を  $p2\text{-atm}^s(p2\text{-ntm}^s, p2\text{-dtm}^s)$  と記す。このとき、

$$P2\text{-ATM}^s(L(n)) = \{T \mid \text{ある } L(n) \text{ 空間限定 } p2\text{-atm}^s M \text{ に対し、} T = T(M)\}$$

$$P2\text{-NTM}^s(L(n)) = \{T \mid \text{ある } L(n) \text{ 空間限定 } p2\text{-ntm}^s M \text{ に対し、} T = T(M)\}$$

$$P2\text{-DTM}^s(L(n)) = \{T \mid \text{ある } L(n) \text{ 空間限定 } p2\text{-dtm}^s M \text{ に対し、} T = T(M)\}$$

と記す。

次に、入力テープが正方形に限定されない一般の  $p2\text{-atm}(p2\text{-ntm}, p2\text{-dtm})$   $M$  を考える。いま、 $L(m, n) : N \times N \rightarrow N$  を2変数  $m$  と  $n$  の関数とする。 $m$  行  $n$  列 ( $m, n \geq 1$ ) のいかなる2次元テープが与えられても、記憶テープの高々  $L(m, n)$  個のコマしか用いないような場合、 $M$  は  $L(m, n)$  空間限定であると言われる。このとき、

$$P2\text{-ATM}(L(m, n)) = \{T \mid \text{ある } L(m, n) \text{ 空間限定 } p2\text{-atm} M \text{ に対し、} T = T(M)\}$$

$$P2\text{-NTM}(L(m, n)) = \{T \mid \text{ある } L(m, n) \text{ 空間限定 } p2\text{-ntm} M \text{ に対し、} T = T(M)\}$$

$$P2\text{-DTM}(L(m, n)) = \{T \mid \text{ある } L(m, n) \text{ 空間限定 } p2\text{-dtm} M \text{ に対し、} T = T(M)\}$$

と記す。

1ペブル2次元交代性有限オートマトン ( $p2\text{-afa}$ )  $M$  は、記憶テープを用いないような  $p2\text{-atm}$  であり、形式的には6項組  $M = (Q, q_0, U, F, \Sigma, \delta)$  で定義される。ここで、 $Q, q_0, U, F, \Sigma$  は、 $p2\text{-atm}$  の場合と同様な意味を持ち、 $\delta$  は遷移関数で、 $Q \times (\Sigma \cup \{\#\}) \times \{0, 1\}$  から  $P(Q \times \{\text{left}, \text{right}, \text{up}, \text{down}, \text{no move}\})$  への写像である。 $Q - U$  に含まれる状態  $q$  は存在状態とよばれる。遷移関数  $\delta$  は、 $M$  の現在の制御部の状態  $q$ 、入力ヘッドが読んでいるコマ上の記号  $\sigma$ 、入力ヘッドが読んでいるコマ上にペブルが置かれているかどうかの情報  $c$  ( $c = 1$  は、そのコマ上にペブルが置かれていることを表し、 $c = 0$  は、そのコマ上にペブルが置かれていないことを表す) とによって、 $M$  の1ステップの動作を決める関

数である。すなわち、もし

$$(q', m) \in \delta(q, \sigma, c)$$

であれば、 $M$  は 1 ステップの動作で、(1) 制御部の状態を  $q'$  に変え、(2) 入力ヘッドを 1 コマ  $m$  の方向に動かす ( $m = left, right, up, down$  であれば、それぞれ左、右、上、下に動かし、また  $m = no\ move$  であれば、動かさない) ことが可能である。

1 ペブル 2 次元非決定性有限オートマトン ( $p2-nfa$ ) は、存在状態だけを持つような  $p2-afa$  であり、1 ペブル 2 次元決定性有限オートマトン ( $p2-dfa$ ) は、遷移関数  $\delta$  によって決められる 1 ステップの動作が一意的であるような  $p2-afa$  である。

入力テープが正方形に限定されているような  $p2-afa(p2-nfa, p2-dfa)$  を  $p2-afa^s(p2-nfa^s, p2-dfa^s)$  と記す。また、

$$P2-AFA^s = \{T \mid \text{ある } p2-afa^s M \text{ に対し、} T = T(M)\}$$

$$P2-NFA^s = \{T \mid \text{ある } p2-nfa^s M \text{ に対し、} T = T(M)\}$$

$$P2-DFA^s = \{T \mid \text{ある } p2-dfa^s M \text{ に対し、} T = T(M)\}$$

$$P2-AFA = \{T \mid \text{ある } p2-afa M \text{ に対し、} T = T(M)\}$$

$$P2-NFA = \{T \mid \text{ある } p2-nfa M \text{ に対し、} T = T(M)\}$$

$$P2-DFA = \{T \mid \text{ある } p2-dfa M \text{ に対し、} T = T(M)\}$$

と記す。

任意の集合  $S$  に対し、 $|S|$  は、 $S$  の要素数を表す。

### 第3章 決定性と非決定性の関係

まえがきでも述べたように、1次元の記号系列を入力対象とする1ペブル1次元チューリング機械では、使用できる記憶テープの空間量が  $o(\log \log n)$  の場合、非決定性と決定性の言語受理能力は同じである。この節では、正方形テープを入力対象とする場合でも、1ペブル2次元チューリング機械は、使用できる記憶テープの空間量が  $o(\log n)$  のとき、非決定性は決定性より受理能力が強いことを示す。

定理1  $L(n) = o(\log n)$  とする。このとき、 $P2-NFA^s$  には含まれるが、 $P2-DTM^s(L(n))$  には含まれないような正方形テープの集合が存在する。

証明  $T = \{x \in \{0, 1\}^{(2)} \mid \exists n \geq 1 [l_1(x) = l_2(x) = 2n \wedge x[(1, 1), (2n, n)] = x[(1, n + 1), (2n, 2n)]]\}$  (すなわち、 $x$  は正方形であり、しかも  $x$  の左半分と右半分が同一である) とする。本定理を証明するために、

- (1)  $T$  は、ある  $p2-nfa^s$  によって受理される、
- (2)  $T$  は、いかなる  $o(\log n)$  空間限定  $p2-dtm^s$  によっても受理されない、  
ということを示す。

(1) の証明：

$T$  は、次のように動作する  $p2-nfa$   $M$  によって受理される。 $2n$  行  $2n$  列 ( $n \geq 1$ ) なる入力テープ  $x$  が  $M$  へ与えられたとする。 $M$  は、 $x$  の左半分に位置する記号

$$\begin{aligned} &x(1, 1), x(2, 1), \dots, x(2n, 1), \\ &x(1, 2), x(2, 2), \dots, x(2n, 2), \\ &\quad \vdots \\ &x(1, n), x(2, n), \dots, x(2n, n), \end{aligned}$$

が、それぞれ  $x$  の右半分に位置する対応する記号

$$\begin{aligned} &x(1, n + 1), x(2, n + 1), \dots, x(2n, n + 1), \\ &x(1, n + 2), x(2, n + 2), \dots, x(2n, n + 2), \\ &\quad \vdots \\ &x(1, 2n), x(2, 2n), \dots, x(2n, 2n), \end{aligned}$$

と同一であるかどうかをこの順番で調べ、対応する記号がすべて同一であることが確かめられたときのみ、受理状態に入って停止するように動作する。一般に、各  $i, j (1 \leq i \leq 2n, 1 \leq j \leq n)$  に対し、記号  $x(i, j)$  が対応する記号  $x(i, n+j)$  と同一であるかどうかを調べたいときには、M はペブルを記号  $x(i, j)$  上に置く。次に、M はその入力ヘッド H を  $x$  の第  $n+j$  番目の列上に動かそうとする。そうするために、M は H が下境界記号 H にぶつかるまで、H を垂直下方に動かしていく。その後、M は H が上境界記号 # にぶつかるまで、H を 2 コマ上に動かすごとに H を 1 コマ右に動かしていく。H が上境界記号 # にぶつかると、H を 1 コマ下に動かす。このとき、H は  $x$  の第  $n+j$  列の第 1 行上にある。M は、 $x$  の第  $i$  行  $n+j$  列の記号  $x(i, n+j)$  を拾い出すため、 $x$  の第  $n+j$  列の各記号上で、次の 2 つの動作のうちの 1 つを非決定的に選択する。

- (i) (現在の記号は  $x(i, n+j)$  でないと推測して) H を 1 コマ下に動かす。
- (ii) 現在の記号が  $x(i, n+j)$  であると推測する。

上の (i) の動作を選択し続けて、H が下境界記号 # に到達すると、 $x(i, n+j)$  の拾い上げに失敗したとして拒否状態に入って停止する。上の (ii) の動作を選択すると、M は H の読む記号を制御部に記憶し、H を左に動かし続ける。もし H がペブルに出会うことなしに左境界記号 # にぶつかれば、M は  $x(i, n+j)$  の拾い上げに失敗したとして、拒否状態に入って停止する。もし H がどこかでペブルに出会うと、M は制御部に記憶されている記号  $x(i, n+j)$  をペブルの下の記号  $x(i, j)$  と比較する。もしこれらの記号が同一であれば、M は  $x(i, j) = x(i, n+j)$  であることを知り、ペブルを 1 コマ下に動かして、記号  $x(i+1, j)$  が対応する記号  $x(i+1, n+j)$  と同一であるかどうかを調べるための準備をする。そうでなければ、M は  $x$  の左半分と右半分が異なることが分かるので拒否状態に入って停止する。ペブルを  $x$  の第  $n$  列の下境界記号 # まで停止することなしに運ぶことができれば、M は  $x$  の左半分と右半分が同一であることを知り、受理状態に入って停止する。ペブルが  $x$  の第  $n$  列の下境界記号 # に到達したことを知るには、ペブルが下境界記号 # に到達するごとに、ペブルの位置から入力ヘッド H を 2 コマ上に動かすごとに 1 コマ右に動かしていく。このとき、 $x$  の右上隅みに到達することを確かめればよい。以上のように動作する M が T を受理することは明らかである。M の形式的定義を付録 A に示す。

### (2) の証明：

(2) の証明は、文献 [?] の定理 2.2 の証明の考えを用いることができる。逆に、ある  $L(n)$  空間限定  $p2-dtm^s$  M が T を受理すると仮定する。ここで  $L(n) = o(\log n)$  である。Q を M の制御部の状態の集合とする。いま、Q を 2 つの互いに素な部分集合  $Q^+$  と  $Q^-$  に分割する。 $Q^+$  と  $Q^-$  は、それぞれ M が制御部にペブルを持っているときと持っていないときの状態の集合に対応する。M は、入力ヘッドを入力テープの最左上端に置き、 $Q^+$  中の初期状態から動作を開始する。一般性を失うことなしに、M は次の条件 (A) を満足すると仮定する：

(A) M は、境界記号 # を越えて入力テープから外に決してはみ出さない (もちろん、境界記号 # の外部から入力テープに入り込むことはない)。また、M が T 中の入力テー

プを受理するときには、M は右境界記号 # 上で  $Q^+$  の中のある受理状態に入って停止する。

以下に、大きい  $n$  に対し、辺長  $2n$  の正方形テープ上での計算を考える。したがって、M は高々  $L(2n)$  個の記憶テープのコマを用いる。各  $n \geq 1$  に対し、 $S(n)$  を、記憶テープの高々  $L(2n)$  個のコマを用いる M の可能な記憶状態からなる集合とする。ここで、記憶状態とは、(i) 記憶テープの内容、(ii) 記憶テープの非空白部分の内容での記憶ヘッドの位置、(iii) 制御部の状態、の組み合わせである。いま、

$$S^+(n) \stackrel{\text{def}}{=} \{s \in S(n) | s \text{ の状態成分が } Q^+ \text{ に含まれる} \}$$

$$S^-(n) \stackrel{\text{def}}{=} \{s \in S(n) | s \text{ の状態成分が } Q^- \text{ に含まれる} \}$$

とする。 $S(n) = S^+(n) \cup S^-(n)$  である。明かに、 $s^+(n) \stackrel{\text{def}}{=} |S^+(n)| = O(t^{L(2n)})$ ,  $s^-(n) \stackrel{\text{def}}{=} |S^-(n)| = O(t^{L(2n)})$  である。ここで、 $t$  は M のみに依存するある定数である。 $x$  を M への入力テープの左半分または右半分となると仮定される  $\{0, 1\}$  上の任意の  $2n$  行  $n$  列のテープ (これを  $(2n, n)$ -block とよぶ) とし、 $x(\#)$  を  $x$  の上辺と下辺に境界記号 # をつけたものとする。すなわち、 $x(\#)$  は、 $l_1(x(\#)) = 2n+2$ ,  $l_2(x(\#)) = n$ ,  $x(\#)[(2, 1), (2n+1, n)] = x$  で、しかも各  $k(1 \leq k \leq n)$  に対し  $x(\#)(1, k) = x(\#)(2n+2, k) = \#$  であるような  $\{0, 1, \#\}$  上の 2 次元テープである。上述の条件 (A) から、 $x(\#)$  への入り口点 (或いは、 $x(\#)$  からの出口点) は、 $x(\#)$  の左辺或いは右辺であることに注意する。 $PT(n)$  を、 $x(\#)$  の左辺あるいは右辺への入力点 (または  $x(\#)$  の左辺あるいは右辺からの出口点) の集合とする。 $|PT(n)| = 2(2n+2) = 4(n+1)$  である。M のペブルは、この  $x(\#)$  上に置かれていないと仮定する。このとき、M と  $x$  で決まる、 $S^-(n) \times PT(n)$  から  $(S^-(n) \times PT(n)) \cup \{l\}$  への写像  $M_x$  を次のように定義する ( $l$  は新しい記号) :

- $M_x(s, p) = (s', p') \Leftrightarrow$  M が記憶状態  $s$  で入口点  $p$  から  $x(\#)$  に入ってくるときに、M が最終的に記憶状態  $s'$  で出口点  $p'$  から  $x(\#)$  を出ていくような M の動作系列が存在する。
- $M_x(s, p) = l \Leftrightarrow$  M が記憶状態  $s$  で入口点  $p$  から  $x(\#)$  に入ってくるときに、M が  $x(\#)$  から決して出ることがないような M の動作系列が存在する。

2 つの  $(2n, n)$ -block  $x, y$  は、各  $(s, p) \in S^-(n) \times PT(n)$  に対し  $M_x(s, p) = M_y(s, p)$  であるならば (すなわち、写像  $M_x$  と  $M_y$  が同一であるならば)、 $M^-$ -同値 であると言う。明らかに、 $M^-$ -同値性は、 $(2n, n)$ -block 上での同値関係である。 $2^{2n \times n} = 2^{2n^2}$  個の  $(2n, n)$ -block が存在する。これらの  $2^{2n^2}$  個の  $(2n, n)$ -block から成る集合を  $M^-$ -同値関係で類別すると、高々

$$e(n) = |(S^-(n) \times PT(n)) \cup \{l\}|^{S^-(n) \times PT(n)} = (4(n+1)s^-(n) + 1)^{4(n+1)s^-(n)}$$

個の  $M^-$ -同値類が存在する。 $P(n)$  を  $(2n, n)$ -block からなる最も大きな 1 つの  $M^-$ -同値類とする。このとき、

$$|P(n)| \geq 2^{2n^2} / e(n) \quad (3.1)$$

が成り立つ。

各  $xy$  に対して ( $x$  と  $y$  は  $(2n, n)$  - block)、

- $comp(xy) \stackrel{\text{def}}{=} M$  の  $xy$  上での計算、
- $cross(xy) \stackrel{\text{def}}{=} comp(xy)$  において、 $M$  が  $x(\#)$  と  $y(\#)$  の間の境界を、左から右または右から左へ横切るときの (i) 記憶状態と (ii) 入力ヘッドの横切る位置の対の系列
- $pebble-cross(xy) \stackrel{\text{def}}{=} comp(xy)$  において、 $M$  が  $x(\#)$  と  $y(\#)$  の間の境界を、制御部の中にペブルを持って、左から右または右から左へ横切るときの (i) ( $S^+(n)$  に含まれる) 記憶状態と (ii) 入力ヘッドの横切る位置の対の系列

とする。  $pebble-cross(xy)$  は、  $cross(xy)$  の部分系列である。

$cross(xy) = c_1 c_2 \cdots c_i \cdots$  における各 (記憶状態と入力ヘッドの横切る位置の) 対  $c_i$  に対して、

- $b-comp(xy, c_i) \stackrel{\text{def}}{=} comp(xy)$  の始めから、 $M$  が対  $c_i$  で  $x(\#)$  と  $y(\#)$  の間の境界を横切る瞬間までの  $comp(xy)$  の部分計算、
- $e-comp(xy, c_i) \stackrel{\text{def}}{=} M$  が対  $c_i$  で  $x(\#)$  と  $y(\#)$  の間の境界を横切った後の  $comp(xy)$  の部分計算

とする。また、  $cross(xy) = c_1 c_2 \cdots c_i \cdots$  における任意の対  $c_i$  と  $c_j$  ( $i < j$ ) に対して、

- $comp(xy, c_i, c_j) \stackrel{\text{def}}{=} M$  が対  $c_i$  で  $x(\#)$  と  $y(\#)$  の間の境界を横切った瞬間から、 $M$  が対  $c_j$  でその境界を再び横切る瞬間までの  $comp(xy)$  の部分計算

とする。このとき、次の補題が成り立たなければならないことが示される。

(補題 1)  $P(n)$  の中の任意の 2 つの異なる  $(2n, n)$ -block  $x, y$  に対して

$$pebble-cross(xx) \neq pebble-cross(yy)$$

(証明:  $x \neq y$  であるにもかかわらず、  $pebble-cross(xx) = pebble-cross(yy)$  と仮定する。 $xx$  と  $yy$  は、両方とも  $T$  に含まれる。したがって、両方とも  $M$  によって受理される。それゆえに、以前に述べられた条件 (A) から、一般性を失うことなしに、ある奇数  $k \geq 1$  に対して、

$$(i) \text{ pebble-cross}(xx) = \text{pebble-cross}(yy) = c_1 c_2 \cdots c_k$$

(各対  $c_i$  の記憶状態成分は  $S^+(n)$  に含まれる),

$$(ii) \text{ cross}(xx) = c_{01}^x c_{02}^x \cdots c_{0i_0}^x c_1 c_{11}^x c_{12}^x \cdots c_{1i_1}^x c_2 c_{21}^x c_{22}^x \cdots c_{2i_2}^x c_3 \cdots c_k c_{k1}^x c_{k2}^x \cdots c_{ki_k}^x$$

( $i_0, i_1, \dots, i_k \geq 0$ , また各対  $c_{ij}^x$  の記憶状態成分は  $S^-(n)$  に含まれる),

$$(iii) \text{ cross}(yy) = c_{01}^y c_{02}^y \cdots c_{0j_0}^y c_1 c_{11}^y c_{12}^y \cdots c_{1j_1}^y c_2 c_{21}^y c_{22}^y \cdots c_{2j_2}^y c_3 \cdots c_k c_{k1}^y c_{k2}^y \cdots c_{kj_k}^y$$

( $j_0, j_1, \dots, j_k \geq 0$ , また各対  $c_{ij}^y$  の記憶状態成分  $S^-(n)$  に含まれる)

であり、しかも  $M$  は入力  $yy$  において、ある右境界記号  $\#$  上で、ある受理状態  $q_a \in Q^+$  に入る。各  $w \in \{x, y\}$  に対して、

(i)  $b\text{-comp}(ww, c_1)$  において、また各偶数  $i (2 \leq i \leq k-1)$  に対する  $\text{comp}(ww, c_i, c_{i+1})$  において、ペブルは左の  $w(\#)$  上にあり、

(ii) 各奇数  $i (1 \leq i \leq k-2)$  に対する  $\text{comp}(ww, c_i, c_{i+1})$  において、また  $e\text{-comp}(ww, c_k)$  において、ペブルは右の  $w(\#)$  上にある

ということに注意。

$x$  と  $y$  は  $M^-$ -同値であるので、次の (i) から (iv) を満たすような  $M$  の  $xy$  上での計算  $\text{comp}(xy)$  を構成することができることが分かる。

$$(i) \text{ cross}(xy) = c_{01}^x c_{02}^x \cdots c_{0i_0}^x c_1 c_{11}^y c_{12}^y \cdots c_{1j_1}^y c_2 c_{21}^x c_{22}^x \cdots c_{2i_2}^x c_3 c_{31}^y c_{32}^y \cdots c_{3j_3}^y c_4 \cdots c_k c_{k1}^y c_{k2}^y \cdots c_{kj_k}^y,$$

$$(ii) \text{ pebble-cross}(xy) = c_1 c_2 \cdots c_k,$$

$$(iii) b\text{-comp}(xx, c_{01}^x) = b\text{-comp}(xy, c_{01}^x)$$

$$[b\text{-comp}(xx, c_1) = b\text{-comp}(xy, c_1) (i_0 = 0 \text{ のとき})],$$

$$(iv) e\text{-comp}(yy, c_{kj_k}^y) = e\text{-comp}(xy, c_{kj_k}^y)$$

$$[e\text{-comp}(yy, c_k) = e\text{-comp}(xy, c_k) (j_k = 0 \text{ のとき})].$$

( 図 2 を見よ。図 2 では、

$$b\text{-comp}(xy, c_{01}^x) = b\text{-comp}(xx, c_{01}^x),$$

$$\text{comp}(xy, c_{02}^x, c_1) = \text{comp}(xx, c_{02}^x, c_1),$$

$$\text{comp}(xy, c_1, c_{11}^y) = \text{comp}(yy, c_1, c_{11}^y),$$

$$\text{comp}(xy, c_{12}^y, c_2) = \text{comp}(yy, c_{12}^y, c_2),$$

$$\text{comp}(xy, c_{22}^x, c_3) = \text{comp}(xx, c_{22}^x, c_3),$$

$$\text{comp}(xy, c_3, c_{31}^y) = \text{comp}(yy, c_3, c_{31}^y), \text{ しかも}$$

$$e\text{-comp}(xy, c_{32}^y) = e\text{-comp}(yy, c_{32}^y)$$

であることに注意せよ。一方、 $\text{comp}(xy, c_{01}^x, c_{02}^x)$ ,  $\text{comp}(xy, c_{11}^y, c_{12}^y)$ ,  $\text{comp}(xy, c_{21}^x, c_{22}^x)$ , な



らびに  $comp(xy, c_{31}^y, c_{32}^y)$  は、 $x$  と  $y$  の  $M^-$ -同値性から生じ、それぞれ  $comp(xx, c_{01}^x, c_{02}^x)$ ,  $comp(yy, c_{11}^y, c_{12}^y)$ ,  $comp(xx, c_{21}^x, c_{22}^x)$ , ならびに  $comp(yy, c_{31}^y, c_{32}^y)$  と必ずしも同一である必要はない。この同一性は、我々の議論にとっては、問題ではない。) )

明らかに、 $comp(xy)$  は、 $M$  が右境界記号  $\#$  上で受理状態  $q_a \in Q^+$  に入って終了する。それゆえに、 $xy$  はまた  $M$  によって受理されることになる。このことは  $xy$  が  $T$  には含まれないという事実に矛盾する。これで、補題 1 の証明は終了する。) )

$M$  は決定性であるので、各  $(2n, n)$ -block  $x$  に対し、 $pebble-cross(xx)$  の中には、(i) ( $S^+(n)$  に含まれる) 記憶状態と (ii) 入力ヘッドの横切る位置との対で同一なものは高々 2 回 (1 回目は、 $M$  が左側の  $x(\#)$  と右側の  $x(\#)$  の間の境界を左から右 (或いは右から左) に横切る場合であり、2 回目は  $M$  がその境界を右から左 (或いは左から右) に横切る場合である) 現れることが分かる。なぜならば、さもないとすると  $comp(xx)$  は無限ループに入り、 $xx$  が受理されなくなってしまうからである。それ故に、各  $(2n, n)$ -block  $x$  に対して、 $pebblem-cross(xx)$  の長さは、 $2(2n+2)s^+(n)$  によって押さえられる。  $n \gg 1$  に対し、

$$PEBBLE-CROSS(n) = \{pebble-cross(xx) | x \text{ は、} (2n, n)\text{-block である} \}$$

とすると、上で観察したことから、

$$|PEBBLE-CROSS(n)| \leq ((2n+2)s^+(n))^{2(2n+2)s^+(n)} \quad (3.2)$$

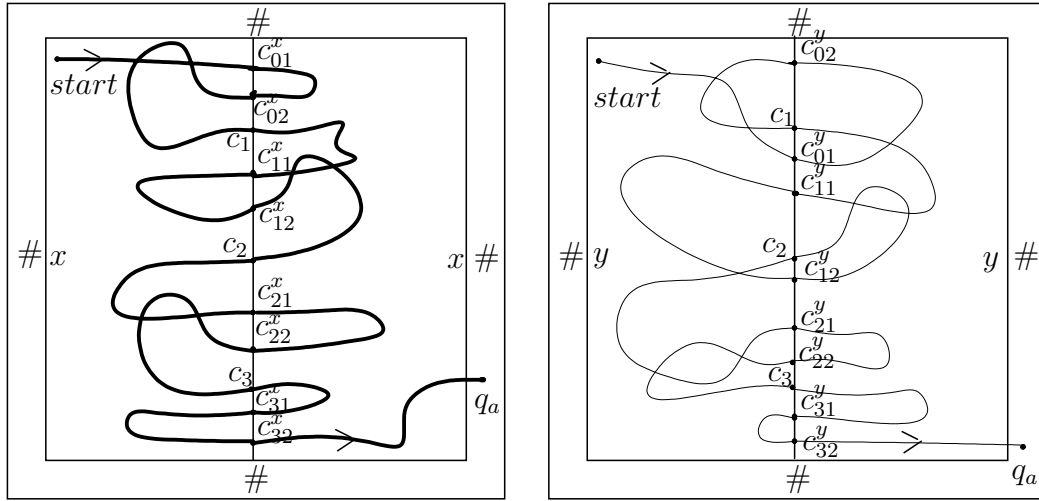
が成り立つ。 $L(n) = o(\log n)$  であるので、式 (3.1) と式 (3.2) から、大きな  $n$  に対しては、 $|P(n)| \gg |PEBBLE-CROSS(n)|$  が成り立つことが示される〔付録 B〕。したがって、 $pebble-cross(xx) = pebble-cross(yy)$  であるような  $P(n)$  に含まれる異なる  $(2n, n)$ -block  $x, y$  が存在しなければならない。このことは、上の補題 1 に矛盾する。これで、(2) の証明を終わる。

定理 1 より、次の系が得られる。

系 1 .  $L(n) = o(\log n)$  のとき、

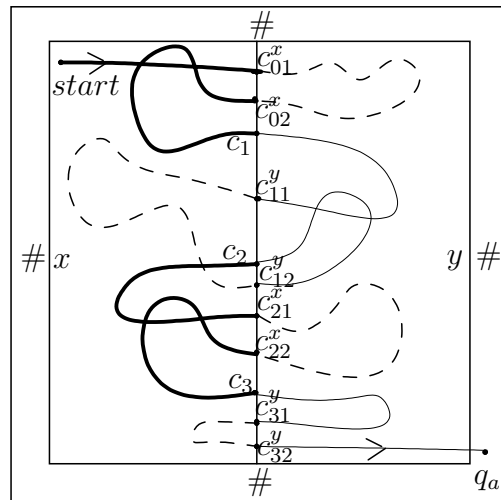
$$P2-DTM^s(L(n)) \subset P2-NTM^s(L(n))$$

系 2 .  $P2-DFA^s \subset P2-NFA^s$



(a)  $comp(xx)$

(b)  $comp(yy)$



(c)  $comp(xy)$

図 3.1: 補題 1 の証明の説明図。

### 3.1 非決定性と交代性の関係

使用できる記憶テープの空間量が  $o(\log n)$  のとき、 $p2-atm^s$  は  $p2-ntm^s$  より受理能力が強いことが予想されるが、その証明は分かっていない。しかし、一般の2次元テープを入力対象とする場合には、次の定理が成り立つ。

定理 2.  $L(m, n) = f(m) + g(n)$  で、 $f(m) = o(\log m)$  かつ  $g(n) = o(\log n)$  とする。このとき、 $P2-AFA$  には含まれるが、 $P2-NTM(L(m, n))$  には含まれないような集合が存在する。

証明. 正の各整数  $m \geq 1$  に対し、 $V(m) = \{x_1c_1x_2c_2 \dots x_m c_m \mid \forall i(1 \leq i \leq m)[x_i \in \{0, 1\}^{(m,m)} \wedge c_i \in \{2\}^{(m,1)}]\}$  とし、 $T_1 = \{xx \mid \exists m \geq 1[x \in V(m)]\}$  とする。本定理を証明するために、

- (1)  $T_1$  は、ある  $p2-afa$  によって受理される、
  - (2)  $T_1$  は、いかなる  $f(m)+g(n)$  空間限定  $p2-ntm$   $M$  ( $f(m) = o(\log m)$  で  $g(n) = o(\log n)$ ) によっても受理されない、
- ということを示す。

(1) の証明 :

$T_1$  は、次のように動作する  $p2-afa$   $M$  によって受理される。 $M$  に次のような形の入力テープ  $w$  が与えられたとする。

$$w = x_1c_1x_2c_2 \dots x_k c_k$$

ここで、 $k \geq 1$  で、各  $i(1 \leq i \leq k)$  に対し、 $x_i \in \{0, 1\}^{(m,m)}$ 、 $c_i \in \{2\}^{(m,1)}$  である (但し、 $m \geq 1$ )。上の形以外の入力テープは、 $M$  によって容易に拒否される。

$M$  はまず  $k = 2m$  であることを次のようにチェックする :

- (1-1) 入力テープ  $w$  の1行1列の位置から出発して、入力ヘッド  $H$  を、記号 '2' に会うまで、右に動かしていく。記号 2 に会えると、ステップ (1-2) へ。
- (1-2)  $H$  を1コマ下に動かし、ステップ (1-3) へ。
- (1-3)  $H$  の読む記号が 2 の場合は、ステップ (1-4) へ。 $H$  の読む記号が下境界記号 # の場合は、ステップ (1-5) へ。
- (1-4)  $H$  を1コマ右に動かす。このとき、
  - $H$  が記号 0 か 1 を読めば、次の新たな記号 '2' に会うまで、 $H$  を右に動かしていき、記号 2 に会えるとステップ (1-2) へもどる。
  - $H$  が右境界記号 # を読めば、 $k \neq 2m$  であることを知り、 $M$  は拒否状態に入る。

(1-5) H を 1 コマ上に動かし、ステップ (1-6) へ。

(1-6) H の読む記号が 2 の場合は、ステップ (1-7) へ。H の読む記号が上境界記号 # の場合は、ステップ (1-8) へ。

(1-7) H を 1 コマ右に動かす。このとき、

- H が記号 0 か 1 を読めば、次の新たな記号 '2' に会うまで、H を右に動かしていき、記号 2 に会おうと、ステップ (1-5) へもどる。
- H が右境界記号 # を読めば、 $k \neq 2m$  であることを知り、M は拒否状態に入る。

(1-8) H を 1 コマ下に動かし、次に H を 1 コマ右に動かす。このとき、

- H が記号 0 か 1 を読めば、 $k \neq 2m$  であることを知り、M は拒否状態に入る。
- H が右境界記号 # を読めば、M は  $k = 2m$  であることを知る。

以上のように、M が  $k = 2m$  であることの確認に成功すると、今度は、すべての  $i (1 \leq i \leq m)$  に対し、 $x_i = x_{m+i}$  であるかどうかを、M は  $i$  の小さい順に調べていく。いま、 $x_i = x_{m+i}$  であるかどうかを調べたいとする。このとき、たとえば、 $x_i$  の  $r$  行  $j$  列の記号  $x_i(r, j)$  と対応する  $x_{m+i}$  の  $r$  行  $j$  列の記号  $x_{m+i}(r, j)$  が同一であるかどうかを調べるのに、M は次のように動作する。M はまず、ペブルを記号  $x_i(r, j)$  上に置く。次に M は入力ヘッドを  $x_{m+i}$  の第  $j$  列上に動かそうとする。そのため、M は、次のようにして、H を  $x_{m+i}$  上に移動させる：

(2-1) H を  $x_i$  の第 1 行上で、記号 2 に会うまで、右に動かしていき、記号 2 に会おうと、ステップ (2-2) へ。

(2-2) H を 1 コマ下に動かし、ステップ (2-3) へ。

(2-3) H の読む記号が 2 の場合は、ステップ (2-4) へ。H の読む記号が下境界記号 # の場合は、ステップ (2-5) へ。

(2-4) H を、次の新たな記号 2 に会うまで、H を右に動かしていき、記号 2 に会おうと、ステップ (2-2) へもどる。

(2-5) H を 1 コマ上に、次に 1 コマ右に動かす。

ステップ (2-5) が終了すると、M は入力ヘッド H を  $x_{m+i}(m, 1)$  上に移動したことになる。ステップ (2-5) が終了すると、次に M は、 $x_{m+i}$  の第  $j$  列を推測するため、 $x_{m+i}$  の第  $m$  行上で右に動いて行き、第  $m$  行の各記号上で、存在状態に入って、次の 2 つの動作のうちの一つを非決定的に選択する：

- (i) (現在の列が  $x_{m+i}$  の第  $j$  列でないと推測し) 右に 1 コマ入力ヘッド H を動かす。

(ii) 現在の列が  $x_{m+i}$  の第  $j$  列であると推測する。

上の (i) の動作を選択し続けて、記号 2 の列に到達すると、M は拒否状態に入って停止する。上の (ii) の動作を選択すると、M は全称状態に入って 2 つの機械 ( ブランチ )  $p$  と  $p_i(r, j)$  に分かれる。、  $p_i(r, j)$  は、現在の列が本当に  $x_{m+i}$  の第  $j$  列であるかどうかを調べ、そうであれば受理状態に入り、まもなければ拒否状態に入って停止する。すなわち、 $p_i(r, j)$  は次のように動作する :

(3-1) 入力ヘッド H を左上方に対角的に動かして行く。このとき、

- H が上境界記号 # にぶつかれば、H を 1 コマ左に動かして ( この時点で、H はステップ (3-1) を実行する直前に比べ、 $m+1$  列左に移動していることに注意 ) ステップ (3-2) へ。
- H が左境界記号 # にぶつかれば、拒否状態に入って停止する。

(3-2) 入力ヘッド H を垂直下方に動かして行く。この途中にペブルにぶつかれば、 $p_i(r, j)$  は受理状態に入って停止する。ペブルにぶつかることなしに下境界記号 # にぶつかれば、H を 1 コマ上に動かして、ステップ (3-1) へ。

一方、ブランチ  $p$  は、現在の列が  $x_{m+i}$  の第  $j$  列であると仮定して ( この仮定が正しいかどうかは、上述のように  $p_i(r, j)$  によって調べられる ) 現在の列を垂直上方に動いて行き、現在の列の各記号上で存在状態に入って、次の 2 つの動作のうちの 1 つを非決定的に選択する :

(iii) ( 現在の行が  $x_{m+i}$  の第  $r$  行でないと推測して ) 上に 1 コマ入力ヘッド H を動かす。

(iv) 現在の行が  $x_{m+i}$  の第  $r$  列であると推測する。

上の (iii) の動作を選択し続けて、上境界記号 # に到達すると、 $p$  は拒否状態に入って停止する。上の (iv) の動作を選択すると、 $p$  は入力ヘッドが現在読んでいる記号 (  $x_{m+i}(r, j)$  であることを期待している ) を有限制御部に記憶し、入力ヘッドを左に動かし続ける。このとき、

- 入力ヘッドがペブルにぶつかれば、 $p$  は有限制御部に記憶されている記号をペブルの下の記号  $x_i(r, j)$  と比較し、これらの記号が同一であれば、 $p$  は  $x_i(r, j) = x_{m+i}(r, j)$  であると判断し、ペブルを 1 コマ下に動かして、今度は記号  $x_i(r+1, j)$  が対応する  $x_{m+i}(r+1, j)$  と同一であるかどうかを調べるための準備をする。さもなければ、 $p$  は  $x_i \neq x_{m+i}$  であると判断し、拒否状態に入って停止する。
- 入力ヘッドがペブルにぶつかることなしに左境界記号 # にぶつかれば、 $p$  は (  $r$  の推測に失敗したとして ) 拒否状態に入って停止する。

以上のように動作するとき、M がペブルを  $x_{m+1}$  の 1 行 1 列の位置まで運ぶことができおれば、M は全ての  $i(1 \leq i \leq m)$  に対し  $x_i = x_{m+i}$  であると判断し、受理状態に入って停止する。M がペブルを  $x_{m+1}$  の 1 行 1 列の位置まで運ぶことができているということは、 $x_{m+1}$  の 1 行 1 列の位置にペブルを運んでくる事が出来た後、この位置から、 $i = m + 1$  のときの上述のステップ (2-1) から (2-4) を繰り返していくとき、途中で右境界記号 # にぶつかることを知るにより確認出来る。

以上のように動作する M が、 $T_1$  を受理することは容易に確かめられる。M の形式的定義を付録 C に示す。

(2) の証明 :

ある  $L(m, n)$  空間限定  $p2-ntm$  M が  $T_1$  を受理すると仮定する。ここで  $L(m, n) = f(m) + g(n)$ ,  $f(m) = o(\log m)$ ,  $g(n) = o(\log n)$  である。Q を M の制御部の状態の集合とする。いま、Q を 2 つの互いに素な部分集合  $Q^+$  と  $Q^-$  に分割する。 $Q^+$  と  $Q^-$  は、それぞれ M が制御部にペブルを持っているときと持っていないときの状態の集合に対応する。M は、入力ヘッドを入力テープの最左上端に置き、 $Q^+$  の中の初期状態から動作を開始する。一般性を失うことなしに、M は定理 1 の証明中で述べた条件 (A) を満足すると仮定する。

各  $m \geq 1$  に対し、 $W(m) = \{xx|x \in V(m)\}$  とする。以下に、大きい正の整数  $m$  に対して、 $W(m)$  の中の入力テープ上での M の計算を考える。 $W(m)$  の中の各テープの行数は  $m$  で、列数は  $2(m+1)m \stackrel{\text{def}}{=} n(m)$  であるので、M は記憶テープのコマを高々  $L(m, n(m))$  個用いる。各  $m \geq 1$  に対し、 $S(m)$  を記憶テープの高々  $L(m, n(m))$  個のコマを用いる M の可能な記憶状態からなる集合とし、

$$S^+(m) = \{s \in S(m) | s \text{ の状態成分が } Q^+ \text{ に含まれる} \}$$

$$S^-(m) = \{s \in S(m) | s \text{ の状態成分が } Q^- \text{ に含まれる} \}$$

とする。 $S(m) = S^+(m) \cup S^-(m)$  である。明らかに、 $s^+(m) \stackrel{\text{def}}{=} |S^+(m)| = O(t^{L(m, n(m))})$ ,  $s^-(m) \stackrel{\text{def}}{=} |S^-(m)| = O(t^{L(m, n(m))})$  である。ここで、 $t$  は M のみに依存するある定数である。

$x$  を M への ( $W(m)$  に含まれる) 入力テープの左半分または右半分となると仮定される  $V(m)$  の中の任意のテープとし、 $x$  の上辺と下辺に境界記号 # を付けたものを  $x(\#)$  と記す。また、 $PT(m)$  を  $x(\#)$  の左辺あるいは右辺からの入口点 (出口点) の集合とすると、 $|PT(m)| = 2(m+2)$  である。M のペブルはこの  $x(\#)$  上に置かれていないと仮定する。M と  $x$  とで決まる、 $S^-(m) \times PT(m)$  から  $(S^-(m) \times PT(m)) \cup \{l\}$  のべき集合への写像  $M_x$  を次のように定義する ( $l$  は新しい記号):

- $(s', p') \in M_x(s, p) \Leftrightarrow$  M が記憶状態  $s$  で入口点  $p$  から  $x(\#)$  に入ってくるときに、M が最終的に記憶状態  $s'$  で出口点  $p'$  から  $x(\#)$  を出ていくような M の動作系列が存在する。

- $l \in M_x(s, p) \Leftrightarrow M$  が記憶状態  $s$  で入口点  $p$  から  $x(\#)$  に入ってくるときに、 $M$  が  $x(\#)$  から決して出ることがないような  $M$  の動作系列が存在する。

$V(m)$  の中の 2 つのテープ  $x, y$  は、写像  $M_x$  と  $M_y$  が同一であるならば  $M^-$ -同値であると言う。明らかに、 $M^-$ -同値性は、 $V(m)$  上での同値関係である。 $|V(m)| = 2^{m^3}$  であり、 $V(m)$  を  $M^-$ -同値関係で類別すると、高々

$$e(m) = (2^{2(m+2)s^-(m)+1})^{2(m+2)s^-(m)}$$

個の  $M^-$ -同値類が存在する。 $P(m)$  を  $V(m)$  の最も大きな 1 つの  $M^-$ -同値類とする。このとき、

$$|P(m)| \geq |V(m)|/e(m) = 2^{m^3}/e(m) \quad (3.3)$$

が成り立つ。

各  $x \in P(m)$  に対し、 $xx$  は  $T_1 \cap W(m)$  に含まれ、したがって  $xx$  は  $M$  によって受理され、 $xx$  の行数は  $m$ 、列数は  $n(m)$  である。したがって、記憶テープのコマを高々  $L(m, n(m))$  個用いるような  $xx$  上の  $M$  の少なくとも一つの受理計算が存在する。“ $comp(xx)$ ” を、 $xx$  上の  $M$  のループのないそのような一つの固定された受理計算とし、

- $pebble-cross(xx) \stackrel{\text{def}}{=} comp(xx)$  において、 $M$  が左の  $x(\#)$  と右の  $x(\#)$  の間の境界を制御部の中にペブルを持って、左から右または右から左へ横切るときの (i) ( $S^+(m)$  に含まれる) 記憶状態と (ii) 入力ヘッドの横切る位置の対の系列

とする。このとき、次の補題が成り立たなければならないことが示される。

(補題 2)  $P(m)$  の中の任意の 2 つの異なる  $x, y$  に対して

$$pebble-cross(xx) \neq pebble-cross(yy)$$

(証明：  $pebble-cross(xx) = pebble-cross(yy)$  と仮定する。補題 1 の証明と同じ考えを用いることにより、受理計算である  $comp(xx)$  と  $comp(yy)$  とから、記憶テープのコマを高々  $L(m, n(m))$  個用いるような  $M$  の  $xy$  上での受理計算が構成できることが示され、したがって、 $xy$  が  $M$  で受理されることになる。このことは、 $xy$  が  $T_1$  に含まれないという事実と矛盾する。これで、本補題の証明を終わる。)

各  $x \in P(m)$  に対し、 $pebble-cross(xx)$  はループを含まない。したがって、各  $x \in P(m)$  に対し、 $pebble-cross(xx)$  の中には、(i) ( $S^+(m)$  に含まれる) 記憶状態と (ii) 入力ヘッドの横切る位置との対で同一ものは高々 2 回 (1 回目は、 $M$  が左側の  $x(\#)$  と右側の  $x(\#)$  の間の境界を左から右 (或いは右から左) に横切る場合であり、2 回目は  $M$  がその境界を

右から左（或いは左から右）に横切る場合である）現れることが分かる。それ故に、各  $x \in P(m)$  に対して、 $\text{pebble-cross}(xx)$  の長さは、 $2(m+2)s^+(m)$  によって押さえられる。各  $m \gg 1$  に対し、

$$PEBBLE-CROSS(m) = \{\text{pebble-cross}(xx) \mid x \in P(m)\}$$

とすると、上で観察したことから

$$|PEBBLE-CROSS(m)| \leq ((m+2)s^+(m))^{2(m+2)s^+(m)} \quad (3.4)$$

が成り立つ。 $L(m, n) = f(m) + g(n)$ ,  $f(m) = o(\log m)$ ,  $g(n) = o(\log n)$  であるので、式 (4.1) と式 (4.2) から、大きな  $m$  に対しては、 $|P(m)| \gg |PEBBLE-CROSS(m)|$  が成り立つことが示される（付録 B）。したがって、 $\text{pebble-cross}(xx) = \text{pebble-cross}(yy)$  であるような  $P(m)$  に含まれる異なる  $x, y$  が存在しなければならない。このことは、上の補題 2 に矛盾する。これで、(2) の証明を終える。（証明終）

定理 2 より、次の系が得られる。

系 3 .  $L(m, n) = f(m) + g(n)$ ,  $f(m) = o(\log m)$ ,  $g(n) = o(\log n)$  のとき、

$$P2\text{-NTM}(L(m, n)) \subset P2\text{-ATM}(L(m, n))$$

系 4 .  $P2\text{-NFA} \subset P2\text{-AFA}$



## 第4章 まとめ

本論文では、2次元テープを入力対象とする1ペブル2次元チューリング機械を新たに提案し、空間量がどのように制限されると、動作の決定性、非決定性、交代性の違いにより、この機械の受理能力に差が生じるかについて考察し、次の結果を得た。

- (1) 正方形テープを入力対象とする  $o(\log n)$  空間限定1ペブル2次元チューリング機械に対しても、非決定性は決定性より受理能力が強い。
- (2)  $L(m, n) = f(m) + g(n)$  で、 $f(m) = o(\log m)$  かつ  $g(n) = o(\log n)$  とする。このとき、 $L(m, n)$  空間限定1ペブル2次元チューリング機械では、交代性は非決定性より受理能力が強い。

1次元テープ(記号の1次元系列)を入力対象とする1ペブル1次元チューリング機械では、 $o(\log \log n)$  空間限定の場合、決定性と非決定性とは受理能力が同じであることが示されており、また、0空間限定の場合(すなわち、1ペブル1次元有限オートマトンの場合)決定性、非決定性、交代性はすべて同じ受理能力を持つことが示されている。このことと上の(1),(2)の結果から、1次元の場合と2次元の場合とで、異なった状況が生じることが分かり、興味深い。

今後の興味ある課題は、例えば、

- 正方形テープを入力対象とする  $o(\log n)$  空間限定1ペブル2次元チューリング機械では、交代性は非決定性より受理能力が強いかどうかを調べること

などである。

### 謝辞

本研究を進めるにあたり、終始御指導頂いた平石邦彦先生、宋 少秋先生、高島 康裕先生に深く感謝致します。また、常に励まして下さった平石研究室の諸氏に感謝致します。

## 参考文献

- [1] M.Blum and C.Hewitt,"Automata on a 2-dimentional Tape",IEEE Conference Record of the 8th Annual Symposium of Switching and Automata Theory, 1967,pp.155-160.
- [2] K.Inoue, I.Takanami and H.Taniguchi, "Two-dimensional alternating Turing machines", Theoretical Computer Science 27, 1983, pp.61-83.
- [3] A.Ito, K.Inoue, I.Takanami and H.Taniguchi, "Two-dimensional alternating Turing machines with only universal states", Information and Control, Vol.35, Nos.1-3, 1982, 193-221.
- [4] T.Okazaki, L.Zhang, K.Inoue, A.Ito and Y.Wang, "A relationship between two-way deterministic one-counter automata and one-pebble deterministic Turing machines", IEICE TRANS.INF. & SYST.,Vol.E82-SD,No.5,1999, pp.999-1004.
- [5] A.Szepietowski,"Turing Machine with Sublogarithmic Space",Lecture Notes in Computer Science 843,Springer-Verlag,1995.

# 付録A (Tを受理する $p2-nfa$ Mの形式的定義)

形式的には、例えば、 $M$  は、次のように定義される（但し、正方形の入力テープのみが  $M$  に与えられるとし、 $M$  は入力テープが正方形であることを調べる必要はないものとする）。 $M = (Q, q_0, U, F, \Sigma, \delta)$ , ここで

1.  $Q = \{(s, 1), (d, 0), (u, 0), (uu, 0), (u, 1), (r, 0), (r, 1), (check, 0), (0, 0), (1, 0), (accept, 0), (reject, 0)\}$
2.  $q_0 = (s, 1)$
3.  $U = \phi$
4.  $F = \{(accept, 0)\}$
5.  $\Sigma = \{0, 1\}$
6.  $\delta$  は次のように定義される：
  - $\delta((s, 1), 0, 0) = \delta((s, 1), 1, 0) = \{(d, 0), down\}$ （現在読んでいるコマ上にペブルを置いて、1コマ下へ動く。）
  - $\delta((d, 0), 0 \text{ または } 1, 0) = \{(d, 0), down\}$ （下境界記号 # にぶつかるまで下方へ動く。）
  - $\delta((d, 0), \#, 0) = \{(u, 0), up\}$
  - $\delta((u, 0), 0 \text{ または } 1, 0 \text{ または } 1) = \{(uu, 0), up\}$
  - $\delta((uu, 0), 0 \text{ または } 1, 0 \text{ または } 1) = \{(r, 0), right\}$
  - $\delta((r, 0), 0 \text{ または } 1, 0) = \{(u, 0), up\}$
  - （下境界記号 # にぶつかると、上境界記号 # にぶつかるまで、2コマ上に動くごとに、1コマ右に動いていく。）
  - $\delta((u, 0), \#, 0) = \{(check, 0), down\}$ （上境界記号 # にぶつかると、1コマ下に動く。）
  - 各  $a, b \in \{0, 1\}$  に対し、

- $\delta((check, 0), a, 0) = \{((a, 0), left), ((check, 0), down)\}$  (現在 読んでいるコマ上の記号  $a$  を記憶して左方向に移動するか、或いは1 コマ下方へ動くかを選択する。)
- $\delta((a, 0), b, 0) = \{((a, 0), left)\}$  (ペブルに出会うまで左方向へ移動する。)
- $\delta((a, 0), \#, 0) = \{reject, 0\}$  (ペブルに出会うことなしに左境界記号  $\#$  に出会うと、 $(reject, 0)$  状態に入る。)
- $\delta((a, 0), a, 1) = \{((s, 1), down)\}$  (ペブルに出会うと、記憶している記号  $a$  とペブルが置かれているコマ上の記号が同じであれば、ペブルを拾って1 コマ下方へ動く。)
- $a \neq b$  ならば、 $\delta((a, 0), b, 1) = \{reject, 0\}$  (記憶している記号  $a$  とペブルが置かれているコマ上の記号  $b$  が異なれば、 $(reject, 0)$  状態に入る。)
- $\delta((check, 0), \#, 0) = \{reject, 0\}$  (左方向へ動くことを選択することなしに下境界記号  $\#$  にぶつかれば、 $(reject, 0)$  状態に入る。)
- -  $\delta((s, 1), \#, 0) = \{((r, 1), right)\}$
- $\delta((r, 1), \#, 0) = \{((u, 1), up)\}$
- $\delta((u, 1), 0 \text{ または } 1, 0) = \{((u, 1), up)\}$
- $\delta((u, 1), \#, 0) = \{((s, 1), down)\}$   
(ペブルを持って下境界記号  $\#$  にぶつくと、ペブルを次の列の先頭行に移動する。)
- $\delta((uu, 0), \#, 0) = \{reject, 0\}$  (入力テープの辺長が奇数であることを知り、 $(reject, 0)$  状態に入る。)
- $\delta((r, 0), \#, 0) = \{accept, 0\}$  (ペブルを入力テープの左半分の全てのコマ上に置き終わっていることを知り、 $(accept, 0)$  状態に入る。)

# 付録B (大きい $n$ に対し、 $|P(n)| \gg |PEBBLE-CROSS(n)|$ であることの証明)

大きい  $n$  に対し、 $|P(n)| \gg |PEBBLE-CROSS(n)|$  であることを示すには、大きい  $n$  に対し、 $2^{2n^2} \gg e(n) \times |PEBBLE-CROSS(n)|$  が成り立つことを示せばよい。 $L(n) = o(\log n)$  であるから、 $\lim_{n \rightarrow \infty} \frac{L(2n)}{\log 2n} = 0$ 、すなわち  $\lim_{n \rightarrow \infty} \frac{L(2n)}{1 + \log n} = 0$  であり、よって

$$\lim_{n \rightarrow \infty} \frac{L(2n)}{\log n} = 0 \quad (\text{B.1})$$

が成り立つ。ところで、

$$\log \log 2^{2n^2} = 1 + 2 \log n \quad (\text{B.2})$$

である。また、

$$\begin{aligned} & \log\{e(n) \times |PEBBLE-CROSS(n)|\} \\ & \leq \log\{(4(n+1)s^-(n) + 1)^{4(n+1)s^-(n)} \times ((2n+2)s^+(n))^{2(2n+2)s^+(n)}\} \\ & = 4(n+1)s^-(n) \log\{4(n+1)s^-(n) + 1\} + 2(2n+2)s^+(n) \log\{(2n+2)s^+(n)\} \\ & = O(nt^{L(2n)} \log\{nt^{L(2n)}\}) \text{ (なぜならば、} s^-(n) = s^+(n) = O(t^{L(2n)}) \text{ だから)} \\ & \text{であるので,} \end{aligned}$$

$$\log \log\{e(n) \times |PEBBLE-CROSS(n)|\} \leq \log n + \log \log n + O(L(2n)) \quad (\text{B.3})$$

が成り立つ。式 (B.1), (B.2), (B.3) より、十分大きな  $n$  に対し、

$$\log \log 2^{2n^2} \gg \log \log\{e(n) \times |PEBBLE-CROSS(n)|\}$$

が成り立つ。よって、 $2^{2n^2} \gg e(n) \times |PEBBLE-CROSS(n)|$  が成り立つ。

# 付録C ( $T_1$ を受理する $p2\text{-afa}$ の形式的定義)

形式的には、例えば、 $M$  は、次のように定義される。 $M = (Q, q_0, U, F, \Sigma, \delta)$ , ここで

1.  $Q = Q_1 \cup Q_2 \cup Q_3 \cup Q_4 \cup \{(reject, 1), (reject, 0), (accept, 0)\}$ , ここで
  - $Q_1 = \{(s, 1), (d, 1), (u, 1), (r, 1), (d\#, 1), (d_2, 1), (u_2, 1), (r_2, 1), (l, 1)\}$ ,
  - $Q_2 = \{(\searrow, 1), (\searrow_r, 1), (\searrow_d, 1), (\searrow_d^2, 1), (u\#, 1), (r\#, 1), (l_v, 1)\}$ ,
  - $Q_3 = \{(\vee, 1), (\vee_d, 1), (\vee_d^2, 1), (\vee_d^\#, 1), (\vee_u, 1), (\vee_u^2, 1), (\vee_u^\#, 1), (l_m, 1)\}$ ,
  - $Q_4 = \{(p, 1), (u, 0), (d_r, 0), (r, 0), (r', 0), (c_g, 0), (c, 0), (c_l, 0), (c_u, 0), (c_\#, 0), (c_d, 0), (r_g, 0), (0, 0), (1, 0), (u_p, 1), (r_p, 1)\}$
2.  $q_0 = (s, 1)$
3.  $U = \{(c, 0)\}$
4.  $F = \{(accept, 0)\}$
5.  $\Sigma = \{0, 1, 2\}$
6.  $\delta$  は次のように定義される :

- (1) まず、入力テープが

$$x_1c_1x_2c_2 \dots x_kc_k$$

の形をしていることをチェックする (ここで、 $k \geq 1$  で、各  $i(1 \leq i \leq k)$  に対し、 $x_i$  は  $m$  行 ( $m \geq 1$ ) をもつ  $\{0, 1\}$  上の 2次元テープで、 $c_i$  は  $m$  行 1列の  $\{2\}$  上の 2次元テープである)。入力テープがこの形をしていなければ、 $M$  は入力テープを拒否するように動作する。このことをチェックするために、 $M$  は状態集合  $Q_1$  を用いる。 $M$  は、入力テープの第 1列、第 2列、第 3列、 $\dots$ , の順番で調べ、各列においては上から下の方向に順次走査していく。このとき、ペブルを用いる必要はなく、ペブルはいつも制御部に持って移動する。状態遷移関数は次のようになる。

- 各  $a \in \{0, 1\}$  に対し、

- $\delta((s, 1), a, 0) = ((d, 1), down)$
- $\delta((d, 1), a, 0) = ((d, 1), down)$   
( $\{0, 1\}$  上の列であるべきことを上から下へチェックしていく。)
- -  $\delta((s, 1), 2, 0) = ((reject, 1), nomove)$
- $\delta((d, 1), 2, 0) = ((reject, 1), nomove)$   
( $\{0, 1\}$  上の列であるべきであるにもかかわらず、途中で記号 2 が出現したときに、拒否状態に入る。)
- -  $\delta((d, 1), \#, 0) = ((u, 1), up)$
- 各  $a \in \{0, 1\}$  に対し、 $\delta((u, 1), a, 0) = ((u, 1), up)$   
( $\{0, 1\}$  上の列を上から下に読んでいるときに、下境界記号#に出会うと、上向きに向きを変え、上境界記号#に出会うまで上方に進む。)
- -  $\delta((u, 1), \#, 0) = ((r, 1), right)$
- $\delta((r, 1), \#, 0) = ((d_{\#}, 1), down)$   
( $\{0, 1\}$  上の列を下から上へ帰っていくときに、上境界記号#に出会うと、上境界記号上で1コマ右に動き、次に1コマ下に動いて新たな列の先頭のコマに入る。)
- 各  $a \in \{0, 1\}$  に対し、 $\delta((d_{\#}, 1), a, 0) = ((d, 1), down)$   
(再び $\{0, 1\}$  上の列が始まったと判断して、 $\{0, 1\}$  上の列を上から下へ走査していこうとする。)
- -  $\delta((d_{\#}, 1), 2, 0) = ((d_2, 1), down)$
- $\delta((d_2, 1), 2, 0) = ((d_2, 1), down)$   
(記号 2 の列が始まると、その列がすべて記号 2 からなることを上から下にチェックしていく。)
- 各  $a \in \{0, 1\}$  に対し、 $\delta((d_2, 1), a, 0) = ((reject, 1), nomove)$   
(記号 2 の列であるべきであるにもかかわらず、途中で記号 0 あるいは 1 が出現したときに、拒否状態に入る。)
- $\delta((d_{\#}, 1), \#, 0) = ((reject, 1), nomove)$   
(入力テープの最後の列が記号 2 だけから成る列でないことが分かると、拒否状態に入る。)
- -  $\delta((d_2, 1), \#, 0) = ((u_2, 1), up)$
- $\delta((u_2, 1), 2, 0) = ((u_2, 1), up)$   
( $\{2\}$  上の列を上から下に読んでいるときに、下境界記号#に出会うと上向きに向きを変え、上境界記号#に出会うまで上方に進む。)
- -  $\delta((u_2, 1), \#, 0) = ((r_2, 1), right)$
- $\delta((r_2, 1), \#, 0) = ((s, 1), down)$   
( $\{2\}$  上の列を下から上へ帰っていくときに、上境界記号#に出会うと

上境界記号上で1コマ右に動き、次に1コマ下に動いて、新たな $\{0, 1\}$ 上のテープの先頭列の先頭のコマに入る。)

- –  $\delta((s, 1), \#, 0) = ((l, 1), left)$
- 各  $a \in \{0, 1, 2\}$  に対し、 $\delta((l, 1), a, 0) = ((l, 1), left)$
- $\delta((l, 1), \#, 0) = ((\setminus_l, 1), right)$   
 (入力テープが上述のような  $x_1c_1x_2c_2\dots x_kc_k$  の形をしていることが分ると、Mは入力テープの先頭行上を左境界記号#にぶつかるまで左に移動し、左境界記号#にぶつかると1コマ右に移動して入力テープの1行1列のコマ上に移動し、各  $x_i(1 \leq i \leq k)$  が正方形になっているかどうかをチェックするルーチンに入る。)

(2) 次に、 $x_1, x_2, \dots, x_k$  の順番に、各  $x_i$  が正方形であるかどうかをチェックする。このため、各  $x_i$  に対して、 $x_i$  の1行1列の位置から右下方45度の方向に進んでいく。このときに用いられる状態集合が  $Q_2$  で、状態遷移関数は次のようになる(この場合も、ペブルを用いる必要はなく、ペブルはいつも制御部に持って移動する。)

- 各  $a \in \{0, 1\}$  に対し、
  - $\delta((\setminus_l, 1), a, 0) = ((\setminus_r, 1), right)$
  - $\delta((\setminus_r, 1), a, 0) = ((\setminus_d, 1), down)$
  - $\delta((\setminus_d, 1), a, 0) = ((\setminus_r, 1), right)$   
 (記号2の列に出会うまで、右方向45度の方向に進んでいく。)
- –  $\delta((\setminus_r, 1), 2, 0) = ((\setminus_d^2, 1), down)$
- $\delta((\setminus_d^2, 1), \#, 0) = ((u_\#, 1), up)$   
 (記号2の列に出会うと、1コマ下方に移動し、移動したコマ上の記号が下境界記号#であれば、 $x_i$  は正方形であることが分かり、記号2の列に沿って上方に移動する準備をする。)
- –  $\delta((\setminus_d, 1), \#, 0) = ((reject, 1), nomove)$
- $\delta((\setminus_d^2, 1), 2, 0) = ((reject, 1), nomove)$   
 ( $x_i$  は正方形でないことが分かり、拒否状態に入る。)
- –  $\delta((u_\#, 1), 2, 0) = ((u_\#, 1), up)$
- $\delta((u_\#, 1), \#, 0) = ((r_\#, 1), right)$
- $\delta((r_\#, 1), \#, 0) = ((\setminus_l, 1), down)$   
 (記号2の列に沿って上方に移動していき、上境界記号#にぶつかると、上境界記号上で1コマ右に動き、次に1コマ下に動いて、新たな  $x_{i+1}$  の1行1列のコマ上に動き、 $x_{i+1}$  が正方形であるかどうかのチェックに入る。)
- –  $\delta((\setminus_l, 1), 1, \#, 0) = ((l_v, 1), left)$



- 各  $a \in \{0, 1, 2\}$  に対し、 $\delta((l_v, 1), a, 0) = ((l_v, 1), left)$
  - $\delta((l_v, 1), \#, 0) = ((v, 1), right)$   
 (すべての  $x_i$  が正方形であることが分かったら、入力テープの第1行に沿って左の方向に移動し、左境界記号#に出会うと1コマ右に動いて、入力テープの1行1列のコマ上に動き、今度は  $k = 2m$  であるかどうかのチェックルーチンに入る。)
- (3) 入力テープが  $x_1c_1x_2c_2 \dots x_kc_k$  の形をしていて、各  $x_i$  は  $\{0, 1\}$  上の  $m$  行  $m$  列の正方形で、各  $c_i$  は  $\{2\}$  上の  $m$  行1列のテープであることが分かったら、今度は、 $k = 2m$  であることをチェックするためのルーチンに入る。このチェックを行なうために用いる状態集合が  $Q_3$  である。状態遷移関数は次のようになる(このときもペブルを用いる必要はなく、Mはペブルを制御部に持って移動する。)
- 各  $a \in \{0, 1\}$  に対し、
    - $\delta((v, 1), a, 0) = ((v_d, 1), right)$
    - $\delta((v_d, 1), a, 0) = ((v_d, 1), right)$   
 (現在の行上で、記号2の列に出会うまで右方に移動していく。これは、本文中で述べたアルゴリズムのステップ(1-1)に相当する。)
  - $\delta((v_d, 1), 2, 0) = ((v_d^2, 1), down)$   
 (記号2の列に出会うと、1コマ下方に移動する。これは、本文中で述べたアルゴリズムのステップ(1-2)に相当する。)
  - -  $\delta((v_d^2, 1), 2, 0) = ((v_d, 1), right)$
  - $\delta((v_d^2, 1), \#, 0) = ((v_d^\#, 1), up)$   
 ( $k = 2m$  だとしたときに、入力テープの左半分の最後行最後列の記号2に到達したかどうかを調べ、まだ到達していなければ、1コマ右へ動き、到達していることが分かったら(すなわち、下境界記号#を読んでいると)、今度は入力テープの右側に  $m$  個の  $\{0, 1\}$  上の  $m$  行  $m$  列のテープがあるかどうかのチェックに入ろうとする。これは、本文中で述べたアルゴリズムのステップ(1-3)と(1-5)に相当する。)
  - -  $\delta((v_d^\#, 1), 2, 0) = ((v_u, 1), right)$
  - 各  $a \in \{0, 1\}$  に対し、 $\delta((v_u, 1), a, 0) = ((v_u, 1), right)$
  - $\delta((v_u, 1), 2, 0) = ((v_u^2, 1), up)$   
 (現在の行上で、記号2の列に出会うまで右方に移動していく。記号2に出会うと、1コマ上方に移動する。これは、本文中で述べたアルゴリズムのステップ(1-6), (1-7)の一部分とステップ(1-5)に相当する。)
  - -  $\delta((v_u^2, 1), 2, 0) = ((v_u, 1), right)$
  - $\delta((v_u^2, 1), \#, 0) = ((v_u^\#, 1), down)$

(入力テープの右半分の先頭行最後列の記号 2 に到達したかどうかを調べ、まだ到達していなければ、1 コマ右へ動き、到達していることが分かると(すなわち、上境界記号 # を読んでいると) 1 コマ下に動く。これは、本文中で述べたアルゴリズムのステップ (1-6) と (1-7),(1-8) の一部分に相当する。)

- $\delta((\vee_u^\#, 1), 2, 0) = ((\vee_u^\#, 1), right)$ 
  - 各  $a \in \{0, 1\}$  に対し、 $\delta((\vee_u^\#, 1), a, 0) = ((reject, 1), nomove)$   
(これは、本文中のアルゴリズムのステップ (1-8) の  $k \neq 2m$  であることを知った場合に相当する。)
- $\delta((\vee_u^\#, 1), \#, 0) = ((l_m, 1), left)$   
(これは、本文中で述べたアルゴリズムのステップ (1-8) の  $k = 2m$  であることを知った場合に相当する。)
- $\delta((l_m, 1), a, 0) = ((l_m, 1), left)$ 
  - $\delta((l_m, 1), \#, 0) = ((p, 1), right)$   
( $k = 2m$  だと知った後に、入力テープの第 1 行に沿って左方に移動し、左境界記号 # にぶつかると、右に 1 コマ動いて入力テープの第 1 行 1 列の位置にヘッドを持ってきて、今度は、入力テープの左半分と右半分が同一であるかどうかのチェックルーチンに入ろうとする。)
- $\delta((\vee_d, 1), \#, 0) = ((reject, 1), nomove)$   
(これは、本文中のアルゴリズムのステップ (1-4) の  $k \neq 2m$  であることを知った場合に相当する。)
- $\delta((\vee_u, 1), \#, 0) = ((reject, 1), nomove)$   
(これは、本文中のアルゴリズムのステップ (1-7) の  $k \neq 2m$  であることを知った場合に相当する。)

(4)  $k = 2m$  であることを知った後に、今度は、全ての  $i(1 \leq i \leq m)$  に対し  $x_i = x_{i+1}$  であるかどうかを、M は  $i$  の小さい順にチェックしていく。このチェックを行なうために用いる状態集合が  $Q_4$  である。 $Q_4$  に含まれる状態のうち  $(p, 1), (c_g, 0), (c, 0), (r_g, 0)$  は、次のような意味を持つ。

- $(p, 1)$ : 記号 0 か 1 を読む時に、ペブルを置くことを指示する状態。
- $(c_g, 0)$  たとえば、 $x_i$  の  $r$  行  $j$  列の記号  $x_i(r, j)$  と  $x_{m+i}$  の  $r$  行  $j$  列の記号  $x_{m+i}(r, j)$  の一致を調べるとき、 $x_i(r, j)$  の位置にペブルを置き、 $x_{m+i}$  の最後行(第  $m$  行)に移動したときに、 $x_{m+i}$  の最後行上で  $x_{m+i}$  の第  $j$  列を推測するとき に用いられる存在状態。
- $(c, 0)$ : 状態  $(c_g, 0)$  によって推測された  $x_{m+i}$  の第  $j$  列が、本当に  $x_{m+i}$  の第  $j$  列になっているかどうかをチェックするルーチン と、( $j$  が正しく推測されたと仮定して)  $x_{m+i}$  の第  $j$  列の " 第  $r$  行目 " のコマを推測するルーチンに枝分かれするとき に用いられる全称状態。

- $(r_g, 0)$ : ペブルの置かれている行を推測するために用いる存在状態。

状態遷移関数は次のようになる。

- 各  $a \in \{0, 1\}$  に対し、
  - $\delta((p, 1), a, 0) = ((u, 0), up)$
  - $\delta((u, 0), a, 0) = ((u, 0), up)$
  - $\delta((u, 0), \#, 0) = ((d_r, 0), down)$   
(ペブルを置き、次にペブルが置かれた列の先頭行にヘッドを移動する。)
- 各  $a \in \{0, 1\}$  に対し、
  - $\delta((d_r, 0), a, 0) = \delta((d_r, 0), a, 1) = ((r, 0), right)$
  - $\delta((r, 0), a, 0) = ((r, 0), right)$
  - $\delta((r, 0), 2, 0) = ((d_r, 0), down)$
  - $\delta((d_r, 0), 2, 0) = ((r, 0), right)$
  - $\delta((d_r, 0), \#, 0) = ((r', 0), up)$
  - $\delta((r', 0), 2, 0) = ((c_g, 0), right)$   
(第1行上で記号'2'に出会うまで右に移動していく。記号'2'に出会うと1コマ下に動く。このとき、記号'2'を読めば、次の新たな記号'2'に出会うまで右に移動する。以後同様な動作を繰り返す。記号'2'を読んで1コマ下に動くとき、下境界記号#を読めば、Mは1コマ上に、次に1コマ右に移動する。これは、本文中のアルゴリズムのステップ(2-1)~(2-5)に相当する。)
- 各  $a \in \{0, 1\}$  に対し、  $\delta((c_g, 0), a, 0) = \{((c_g, 0), right), ((c, 0), nomove)\}$   
( $x_i$ の第j列にペブルが置かれているとき、現在の列が $x_{m+i}$ の第j列かどうかを推測し、第j列でないと推測したときは、 $((c_g, 0), right)$ を選択し、第j列であると推測したときは、 $((c, 0), nomove)$ を選択する。)
- $\delta((c_g, 0), 2, 0) = ((reject, 0), nomove)$   
(本文中のアルゴリズムの(i)の動作を選択し続けて、記号2に到達する場合に相当する)
- 各  $a \in \{0, 1\}$  に対し、  $\delta((c, 0), a, 0) = \{((c_l, 0), left), ((r_g, 0), nomove)\}$   
( $x_i$ の第j列にペブルが置かれているとき、現在の列が $x_{m+i}$ の第j列であると推測して、2つのブランチに枝分かれする。)
- 各  $a \in \{0, 1, 2\}$  に対し、
  - $\delta((c_l, 0), a, 0) = ((c_u, 0), up)$
  - $\delta((c_l, 0), \#, 0) = ((reject, 0), nomove)$
  - $\delta((c_u, 0), a, 0) = ((c_l, 0), left)$
  - $\delta((c_u, 0), \#, 0) = ((c_\#, 0), left)$

- $\delta((c_{\#}, 0), \#, 0) = ((c_d, 0), down)$
- $\delta((c_d, 0), a, 0) = ((c_d, 0), down)$
- $\delta((c_d, 0), a, 1) = ((accept, 0), nomove)$
- $\delta((c_d, 0), \#, 1) = ((c_u, 0), up)$   
 ( $x_i$  の第  $j$  列にペブルが置かれているとき、現在の列が  $x_{m+i}$  の第  $j$  列であると推測した後、この推測が正しいかどうかを調べる。これは、本文中のアルゴリズムのステップ (3-1),(3-2) に相当する。)
- 各  $a \in \{0, 1\}$  に対し、 $\delta((r_g, 0), a, 0) = \{((r_g, 0), up), ((a, 0), left)\}$   
 (ペブルの置かれている行を推測し、現在の行がペブルの置かれている行でないとは推測すると  $((r_g, 0), up)$  を選択し、現在の行がペブルの置かれている行であると推測すると  $((a, 0), left)$  を選択して現在読んでいる記号  $a$  を制御部に記憶して左に 1 コマ動かす。)
- 各  $a \in \{0, 1\}$ , 各  $b \in \{0, 1, 2\}$  に対し、 $\delta((a, 0), b, 0) = ((a, 0), left)$   
 (ペブルにぶつかるまで記号  $a$  を制御部に記憶して左へ移動する。)
- 各  $a \in \{0, 1\}$ , 各  $b \in \{0, 1\}$  に対し、  
 $\delta((a, 0), b, 1) = ((p, 1), down) \dots a = b$  のとき  
 $= ((reject, 0), nomove) \dots a \neq b$  のとき  
 (ペブルにぶつかると、制御部に記憶されている記号  $a$  とペブルの置かれているコマ上の記号  $b$  を比較し、 $a = b$  であれば、1 コマ下方に動いて、次の位置の記号の比較に入ろうとする。 $a \neq b$  であれば、拒否状態に入る。)
- $\delta((r_g, 0), \#, 0) = ((reject, 0), nomove)$   
 (これは、本文中のアルゴリズムの (iii) の動作を選択し続けて、上境界記号  $\#$  に到達する場合に相当する。)
- 各  $a \in \{0, 1\}$  に対し、 $\delta((a, 0), \#, 0) = ((reject, 0), nomove)$   
 (ペブルの置かれている行の推測に失敗したとき、拒否状態に入る。)
- 各  $a \in \{0, 1\}$  に対し、
  - $\delta((p, 1), \#, 0) = ((u_p, 1), up)$
  - $\delta((u_p, 1), a, 0) = ((u_p, 1), up)$
  - $\delta((u_p, 1), \#, 0) = ((r_p, 1), right)$
  - $\delta((r_p, 1), \#, 0) = ((p, 1), down)$
  - $\delta((p, 1), 2, 0) = ((p, 1), right)$   
 (現在の列上のペブルを下境界記号  $\#$  まで持ってくるのができたことが分かると、右隣の列 (右隣の列が記号 '2' の列のときは、さらに右隣の列) の先頭行にペブルを持ってくる。)
- $\delta((c_g, 0), \#, 0) = ((accept, 0), nomove)$   
 (入力テープの右半分の第 1 行 1 列の位置までペブルを移動することができたことが分かり、入力テープの左半分と右半分が同一であることを知っ

て、受理状態に入る。)

# 付録D (大きい $m$ に対し、 $|P(m)| \gg |PEBBLE-CROSS(m)|$ であることの証明)

大きい  $m$  に対し、 $|P(m)| \gg |PEBBLE-CROSS(m)|$  であることを示すには、大きい  $m$  に対し、 $|V(m)| = 2^{m^3} \gg e(m) \times |PEBBLE-CROSS(m)|$  が成り立つことを示せばよい。 $f(m) = o(\log m)$  であるので、

$$\lim_{m \rightarrow \infty} \frac{f(m)}{\log m} = 0 \quad (\text{D.1})$$

である。また、 $g(n) = o(\log n)$  であるので  $\lim_{n(m) \rightarrow \infty} \frac{g(n(m))}{\log n(m)} = 0$  であり、この式で  $n(m) = 2(m+1)m$  を代入して

$$\lim_{m \rightarrow \infty} \frac{g(n(m))}{\log 2 + \log(m+1) + \log m} = 0$$

が得られる。この式から、

$$\lim_{m \rightarrow \infty} \frac{g(n(m))}{\log m} = 0 \quad (\text{D.2})$$

が得られる。また、

$$\begin{aligned} & \log\{e(m) \times |PEBBLE-CROSS(m)|\} \\ &= \log e(m) + \log|PEBBLE-CROSS(m)| \\ &\leq \log\{2^{2(m+2)s^-(m)+1} 2^{(m+2)s^-(m)}\} + \log\{((m+2)s^+(m))^{2(m+2)s^+(m)}\} \\ &= O(m^2 s^-(m)^2) + O(ms^+(m)(\log m + \log s^+(m))) \\ &= O(m^2 t^{2L(m,n(m))} + mt^{L(m,n(m))}(\log m + L(m,n(m))\log t)) \\ & \text{(なぜならば、} s^-(m) = s^+(m) = O(t^{L(m,n(m))}) \text{だから)} \\ &= O(m^2 b^{L(m,n(m))}) \text{(ここで、} b \text{はある定数)} \end{aligned}$$

であり、よって、 $\log\log\{e(m) \times |PEBBLE-CROSS(m)|\} \leq 2\log m + O(L(m,n(m)))$  が成り立つ。ところで、 $L(m,n(m)) = f(m) + g(n(m))$  であるので、

$$\log\log\{e(m) \times |PEBBLE-CROSS(m)|\} \leq 2\log m + O(f(m) + g(n(m))) \quad (\text{D.3})$$

が得られる。一方、

$$\log \log |V(m)| = 3 \log m \quad (\text{D.4})$$

である。式 (D.1),(D.2),(D.3),(D.4) より、十分大きな  $m$  に対し、

$$\log \log |V(m)| \gg \log \log \{e(m) \times |\text{PEBBLE-CROSS}(m)|\}$$

が成り立つ。よって、 $|V(m)| \gg e(m) \times |\text{PEBBLE-CROSS}(m)|$  が成り立つ。