

万能可逆チューリング機械の一構成法

A construction method of universal reversible Turing machines

安部 崇嗣 ・ 森田 憲一

Takashi Abe and Kenichi Morita

広島大学大学院工学研究科

Graduate School of Engineering, Hiroshima University

概要

任意の可逆チューリング機械を模倣できる万能可逆チューリング機械を構築する問題を研究した。このとき、1次元1テープ万能可逆チューリング機械を構築することは可能であるが、状態数と記号数を少なくすることは非常に困難である。本研究では、より少ない状態数と記号数で万能可逆チューリング機械を構築するために2テープチューリング機械を用いた。この2テープの内の1つは模倣される可逆チューリング機械のテープをそのまま記述し、もう一つのテープは2次元テープにし、模倣される可逆チューリング機械の記述を持たせる方法をとった。これにより、一つのテープが2記号、もう一つのテープが8記号の、9状態(2,8)記号2テープ2次元万能可逆チューリング機械と、二つのテープが共に2記号の、45状態(2,2)記号2テープ2次元万能可逆チューリング機械が得られた。

キーワード : 可逆コンピューティング,
万能チューリング機械

Keywords : reversible computing,
universal Turing machine

1 はじめに

現在の半導体論理素子は2020年頃に微細化の限界に達すると言われており、それ以後は物質の原子・分子レベルの性質を直接論理演算に用いるような素子へ移行すると見られている。可逆

コンピューティングは、物質のミクロな可逆的物理解法則を反映した計算モデルであり、このような方向の研究における一つの鍵となると考えられる。本論文では万能可逆チューリング機械を構成する問題を研究した。通常の(非可逆な)万能チューリング機械の研究はこれまでに多くあり、非常に少ない状態数と記号数を持つものがいくつか示されている[2, 4]。どれほど少ない状態数と記号数を持つ万能可逆チューリング機械が存在するかは、汎用の可逆計算機構を構成する際の基礎となるという点で重要である。

Bennett [1] は、任意の(非可逆な)1テープ2記号チューリング機械が与えられたときに、それと等価な3テープ多記号可逆チューリングが構成できることを示している。同様に、Moritaら [3] は元の機械と等価な1テープ2記号可逆チューリング機械を構成する方法を示している。従って、例えば7状態4記号の万能チューリング機械 [2] にこれらの方法を適用することによって万能可逆チューリング機械を得ることができるが、Bennettの方法だと74状態(4,29,4)記号(つまり3本のテープの記号数がそれぞれ4, 29, 4)、また、Moritaらの方法だと約150,000状態2記号と、非常に複雑なものになってしまう。このため、より直接的な方法でより簡潔な万能可逆チューリング機械を構成することが求められる。しかし直接的な方法を採用しても、1テープの万能可逆チューリング機械を作る場合には、非可逆な万能チューリング機械に比べてかなり複雑になると予想される。そこで本研究では2

テープ 2 次元可逆チューリング機械の枠組みを用いることで状態数と記号数の少ないものを構成するという試みを行った。

万能可逆チューリング機械を構築する際、模倣されるチューリング機械の状態遷移とテープ演算をどのように可逆的に実現するかが一番の問題点となる。これを解決するため、模倣されるチューリング機械を可逆チューリング機械に限定し (Bennett の結果からこの制限により一般性が失われることはない)、動作規則の記述を 2 次元テープ上に与える方法を用いた。2 次元テープの幾何学的性質を用いることにより、状態遷移と演算の実行を少ない状態数と記号数で可逆的に実現でき、その結果、9 状態 (2,8) 記号、および 45 状態 (2,2) 記号の 2 テープ 2 次元万能可逆チューリング機械が得られた。

2 諸定義

定義 2.1 1 テープチューリング機械 T は

$$T = (Q, S, q_0, q_f, s_0, M)$$

によって定義される。各項目は次の通り。

Q ... 有限制御部の内部状態の有限集合

S ... テープ記号の有限集合

$q_0 \in Q$... 有限制御部の初期状態

$q_f \in Q$... 有限制御部の最終状態

$s_0 \in S$... 空白記号

M ... 動作規則の集合

M は $(Q \times S \times S \times Q) \cup (Q \times \{/\} \times \{-, 0, +\} \times Q)$ の部分集合で、動作関数 (動作規則の集合) を表す。-, 0, + はヘッドの移動方向を表し、それぞれ左移動、移動なし、右移動を意味する。 M の各要素は 4 項組と呼ばれ、テープへの操作と有限制御部の状態遷移を記した「命令」である。各命令は、 $[q, s, s', q']$ または $[q, /, d, q']$ の形 ($q, q' \in Q, s, s' \in S, d \in \{-, 0, +\}$) をしている。 $[q, s, s', q']$ はテープ記号の読取りと書換え操作を行う命令で、有限制御部の内部状態が q でテープ記号 s を読んだとき、記号を s' に書換え内部状態を q' にするという動作を表す。 $[q, /, d, q']$ はヘッド位置の移動操作を行う命令で、有限制御

部の内部状態が q のとき、ヘッドを d の方向に移動させ内部状態を q' にするという動作を表す。

次に可逆性を定義する。通常のコピーリング機械の定義では 4 項組ではなく $[q, s, q', s', d]$ というような形の 5 項組が用いられることが多い。しかし、文献 [1] にあるように、チューリング機械の可逆性を定義するには 4 項組を使う方が便利である。

$m_1 = [q_1, b_1, c_1, q'_1], m_2 = [q_2, b_2, c_2, q'_2]$ の二つの 4 項組を考える。

(i) 次の条件が成り立つ時、 m_1 と m_2 は値域が重複するという。

$$q'_1 = q'_2 \text{ かつ } [c_1 = c_2 \text{ または } b_1 = / \text{ または } b_2 = /]$$

(ii) 次の条件が成り立つ時、 m_1 と m_2 は定義域が重複するという。

$$q_1 = q_2 \text{ かつ } [b_1 = b_2 \text{ または } b_1 = / \text{ または } b_2 = /]$$

与えられたチューリング機械において、 M のどの二つの異なる 4 項組も値域が重複しない時に可逆チューリング機械と呼び、どの二つの異なる 4 項組も定義域が重複しない時に決定性チューリング機械と呼ぶ。なお、以下で論じるチューリング機械は決定性のものに限ることとする。

例 2.1 次に与えられる T_1 は (決定性の) 可逆チューリング機械である。

$$T_1 = (\{q_0, q_1, q_2, q'_1, q'_2, q_f\}, \{0, 1\}, q_0, q_f, 0, M)$$

但し M は次のとおり。

$$M = \{[q_0, 0, 0, q'_1], [q_1, 0, 0, q_f], [q_1, 1, 1, q'_2], [q_2, 0, 1, q'_1], [q_2, 1, 0, q'_2], [q'_1, /, +, q_1], [q'_2, /, -, q_2]\}$$

次に、本論文で用いる 2 テープ 2 次元チューリング機械を定義する。これは、1 次元テープと 2 次元テープを 1 つずつ持つモデルである。

定義 2.2 2 テープ 2 次元チューリング機械 T は

$$T = (Q, (S_1, S_2), q_0, q_f, (s_0, s'_0), M)$$

によって定義される。但し、 Q, q_0, q_f は1テープチューリング機械と同様であり、 S_1, S_2 はそれぞれ1次元と2次元テープで使われるテープ記号の集合、 s_0, s'_0 は各テープの空白記号である。4項組集合 M は $(Q \times (S_1 \times S_2) \times (S_1 \times S_2) \times Q) \cup (Q \times \{/, /\} \times \{-, 0, +\} \times \{0, \uparrow, \rightarrow, \downarrow, \leftarrow\}) \times Q) \cup (Q \times (\{/\} \times S_2) \times (\{-, 0, +\} \times S_2) \times Q) \cup (Q \times (S_1 \times \{/\}) \times (S_1 \times \{0, \uparrow, \rightarrow, \downarrow, \leftarrow\}) \times Q)$ の部分集合となる。2テープ2次元チューリング機械に対する可逆性と決定性の定義も1テープの場合と同様である。

3 万能可逆チューリング機械の構築

3.1 9状態(2,8)記号2テープ2次元万能可逆チューリング機械 U_9

任意の2記号1テープ可逆チューリング機械をシミュレートできる、9状態(2,8)記号2テープ2次元万能可逆チューリング機械 U_9 は、

$$U_9 = (\{q_0, q'_0, q_1, q'_1, q_2, q'_2, q_3, q'_3, q_f\}, (\{0, 1\}, \{0, 1, X, Y, L, R, K, F\}), q_0, q_f, (0, 0), M)$$

によって定義される。ただし、 M は次のとおり。

$$M = \{[q_0, [/, 0], [0, 0], q'_0], [q_0, [1, X], [1, X], q'_1], [q_0, [/, 1], [0, 1], q'_2], [q_0, [0, X], [0, X], q'_0], [q_2, [/, 0], [0, 0], q'_2], [q_0, [0, K], [1, K], q'_0], [q_2, [/, 1], [0, 1], q'_3], [q_0, [1, K], [0, K], q'_0], [q_3, [/, 0], [0, 0], q'_3], [q_0, [/, R], [+ , R], q'_0], [q_3, [/, 1], [0, 1], q'_1], [q_0, [/, L], [- , L], q'_0], [q_1, [/, 0], [0, 0], q'_1], [q_0, [0, Y], [0, Y], q'_0], [q_1, [/, 1], [0, 1], q'_0], [q_1, [1, Y], [1, Y], q'_0], [q'_0, [/, /], [0, \downarrow], q_0], [q'_1, [/, /], [0, \leftarrow], q_1], [q'_2, [/, /], [0, \rightarrow], q_2], [q'_3, [/, /], [0, \uparrow], q_3], [q_0, [/, F], [0, F], q_f]\}$$

U_9 の1次元テープは T のテープをそのまま表現する。ヘッドの位置も同じである。 U_9 の2次元テープは T の記述を保持する。以下では、任意の2記号1テープ可逆チューリング機械 T に

対して、 U_9 の2次元テープに与えるべき T の記述を構成する方法を示す。ここでは、例2.1の1テープチューリング機械 T_1 を取り上げてその記述法を例示する。

T の各4項組によって実行されるテープ演算は、「記号読取り」、「記号反転」(0を1に、または1を0に書換える)、「ヘッドの」レフトシフト、「ライトシフト」、の合計4種の演算に分解できる。

ここで注意すべきことは、記号の読取りは一般に「分岐」を伴うことである。つまり、読取った記号が0か1かに依存して次にとる状態が変わる。また、分岐の直後には一般に「合流」が生じる。つまり、他の4項組によって遷移した後の状態と同一になり得る(そうでなければ無際限に多くの状態が必要になってしまう)。しかし、もし T が可逆であれば、遷移後の状態が同一であるような二つの異なる読取り/書換え命令における書換え後の記号は異なるはずである。 U_9 においては、この書換え後の記号を読むことによって可逆的に合流させることができる。 U_9 のこの演算を「逆読み取り」(読取りの逆演算)と呼ぶことにする。

U_9 の2次元テープは0と1以外に、上記5種類の演算と、 T の停止に対応するものの、合計6種類の命令記号(X, R, L, K, Y, F)を持っており、これらを2次元テープ上に適切に配置する。 T の記述は、2次元テープの長方形領域の領域に書き下す。図1の2次元テープは T_1 の記述が書かれている領域だけを示している。 T の各状態(最終状態 q_f を除く)に対して、長方形領域の列を左から順に割り当てる。このとき、記号を読み取る状態には2列を、ヘッドシフト状態には1列を、それぞれ割り当てる。

各命令の役割とそれに対する U_9 の動作を順に説明する。

(1) X : 読み取り命令

記号読取り状態(T_1 では q_1 と q_2) のそれぞれについて、対応すね列に X を置く。 U_9 は、 X を読むと、それと同時に1次元テープの記号も読み取り、0か1かにより分岐を行う。0ならば2次元テープヘッドを下にシフトし、1ならば2次

元テープヘッドを左にシフトする。(後者の場合 U_9 は, さらに, 記号 X の左にある記号 1 を読んで, その下の行にヘッドをシフトする.)

(2) R, L : ライトシフトとレフトシフト命令
ヘッドシフト状態 (T_1 では q'_1 と q'_2) に対しては, それぞれの列に R または L を置く. R を読むと 1 次元テープヘッドを右へシフトする. L を読むと左へシフトする.

(3) K : 記号反転命令
記号を読み取った後, 1 次元テープ記号を反転させる場合 (T_1 では q_2) にはそのための命令 K を置く. U_9 は K を読むと, 1 次元テープ上の記号が 0 ならば 1 に, 1 ならば 0 に書換える.

(4) F : 停止命令
最終状態に遷移する場合 (T_1 では状態 q_1 で 1 次元テープ記号 0 を読んだとき) には停止命令 F を置く.

(5) Y : 逆読み取り命令
 T_1 において, 例えば状態 q_1 で 1 次元テープ記号が 1 の時と, 状態 q_2 で 1 次元テープ記号が 1 の時は, 共に状態 q'_2 になるので, 合流を行う必要がある. 合流地点には Y を置き, この記号と 1 次元テープ上の記号を同時に読むことによって可逆的に合流する.

T の状態遷移は 2 次元テープ上のある列から別の列へ U_9 の 2 次元テープヘッドを移動させることで実現できるが, これは 2 次元テープヘッドの方向転換 (左折) を表す記号 1 を適切に配置することで実現できる. 図 1 は上記の記述法を元に構成した T_1 の記述である. 灰色の部分 T_1 の 4 項組 $[q_1, 1, 1, q'_2]$ の実行に対応する U_9 のヘッドの軌跡を示す.

U_9 は, T の初期ヘッド位置と同じ位置に 1 次元テープヘッドを置き, T の初期状態に対応する列の最上行に 2 次元テープヘッドを置いて動作を開始することにより, T の計算過程を模倣することが出来る. また, U_9 が可逆であることは 4 項組集合から確かめられる.

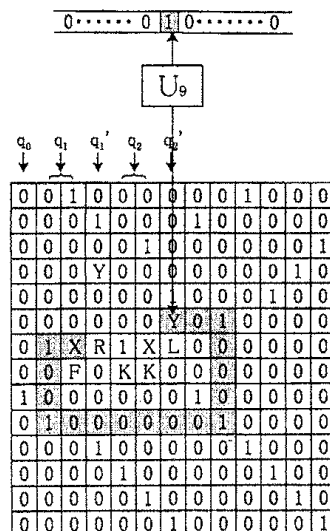


図 1: T_1 を模倣する 9 状態 (2,8) 記号 2 テープ 2 次元万能可逆チューリング機械 U_9

3.2 45 状態 (2,2) 記号 2 テープ 2 次元万能可逆チューリング機械 U_{45}

45 状態 (2,2) 記号 2 テープ 2 次元万能可逆チューリング機械 U_{45} は, 9 状態 (2,8) 記号 2 テープ 2 次元万能可逆チューリング機械の各命令記号を 0,1 の 2 記号で符号化した 2 次元テープを持つ. 非可逆チューリング機械においては任意の数の記号を 0,1 の 2 記号で符号化するのは容易であるが, 可逆チューリング機械においてはこれは自明ではない.

例えば, ある命令記号を [1010] で符号化したとする. この符号を 1 文字ずつ読みながら内部状態で記憶することは可能である. 4 ビットを読み終えた時点で元の記号が復元でき, それに対応する演算が可能となる. しかし演算後, 新たに符号を読んで演算を実行するためには記憶した符号を可逆的に忘却する必要がある (そうでなければ, 状態数が無際限に多く必要となる). そのためここでは, [1010] を反転したビット列 [0101] を元の符号の後ろにつけ, これを読みながら可逆的に忘却する方法を用いた.

U_{45} は次のように定義される.

$$U_{45} = (Q, (\{0, 1\}, \{0, 1\}), q_0, q_f, 0, M)$$

但し Q は次のとおり.

$$Q = \{q_0, q'_0, q_{01}, q'_{01}, q_{010}, q'_{010}, q_{0100}, q'_{0100}, q_{0101},$$

$q'_{0101}, q_{01000}, q'_{01000}, q_{01001}, q'_{01001}, q_{01010},$
 $q'_{01010}, q_{010000}, q'_{010000}, q_{010001}, q'_{010001},$
 $q_{010010}, q'_{010010}, q_{010011}, q'_{010011}, q_{0100101},$
 $q'_{0100101}, q_1, q'_1, q_2, q'_2, q_3, q'_3, q''_{01}, q'''_{01}, q''_{010},$
 $q'''_{010}, q''_{0100}, q'''_{0100}, q''_{0101}, q'''_{0101}, q''_{01000}, q'''_{01000},$
 $q''_{01001}, q'''_{01001}, q_f$

M は以下に示す 4 項組の集合である。なお、各命令に対応する符号の図はスペースの関係で横向きになっているが、実際には縦方向に記号が並んでいる。

X (読取り命令, [10001] で符号化) に対応する 4 項組:

$$\text{X } a_0 \longrightarrow \begin{array}{c} \boxed{10001} \\ \boxed{01100101} \end{array} \longrightarrow a_0$$

$[q_0, [/ , 0], [0, 0], q'_0] [q'_0, [/ , /], [0, \downarrow], q_0]$
 $[q_0, [/ , 1], [0, 1], q'_{01}] [q'_{01}, [/ , /], [0, \downarrow], q_{01}]$
 $[q_{01}, [/ , 0], [0, 0], q'_{010}] [q'_{010}, [/ , /], [0, \downarrow], q_{010}]$
 $[q_{010}, [/ , 0], [0, 0], q'_{0100}] [q'_{0100}, [/ , /], [0, \downarrow], q_{0100}]$
 $[q_{0100}, [/ , 0], [0, 0], q'_{01000}] [q'_{01000}, [/ , /], [0, \downarrow], q_{01000}]$
 $[q_{01000}, [0, 1], [0, 1], q'_{010000}] [q'_{010000}, [/ , /], [0, \downarrow], q_{010000}]$
 $[q_{010000}, [1, 1], [1, 1], q'_{0100001}] [q'_{0100001}, [/ , /], [0, \leftarrow], q_{0100001}]$
 $[q_{0100001}, [/ , 0], [0, 0], q'_{0100000}]$
 $[q'_{0100000}, [/ , 1], [0, 1], q'_{0100000}] [q'_{0100000}, [/ , /], [0, \downarrow], q''_{010000}]$
 $[q''_{010000}, [/ , 0], [0, 0], q'_{0100000}] [q'_{0100000}, [/ , /], [0, \downarrow], q''_{01000}]$
 $[q'_{01000}, [/ , 0], [0, 0], q'_{010000}] [q'_{010000}, [/ , /], [0, \downarrow], q'_{01000}]$
 $[q'_{01000}, [/ , 0], [0, 0], q'_{010000}] [q'_{010000}, [/ , /], [0, \downarrow], q'_{01000}]$
 $[q'_{01000}, [/ , 1], [0, 1], q'_0] [q'_0, [/ , /], [0, \downarrow], q_0]$

Y (逆読み取り命令, [10010] で符号化) に対応する 4 項組:

$$\text{Y } a_0 \longrightarrow \begin{array}{c} \boxed{10010} \\ \boxed{110011} \end{array} \longrightarrow a_0$$

$[q_0, [/ , 0], [0, 0], q'_0] [q'_0, [/ , /], [0, \downarrow], q_0]$
 $[q_0, [/ , 1], [0, 1], q'_{01}] [q'_{01}, [/ , /], [0, \downarrow], q_{01}]$
 $[q_{01}, [/ , 0], [0, 0], q'_{010}] [q'_{010}, [/ , /], [0, \downarrow], q_{010}]$
 $[q_{010}, [/ , 0], [0, 0], q'_{0100}] [q'_{0100}, [/ , /], [0, \downarrow], q_{0100}]$
 $[q_{0100}, [/ , 1], [0, 1], q'_{01001}] [q'_{01001}, [/ , /], [0, \downarrow], q_{01001}]$
 $[q_{01001}, [/ , 0], [0, 0], q'_{010010}] [q'_{010010}, [/ , /], [0, \downarrow], q_{010010}]$
 $[q_{010010}, [/ , 1], [0, 1], q'_{0100101}] [q'_{0100101}, [/ , /], [0, \leftarrow], q_{0100101}]$
 $[q_{0100101}, [1, 1], [0, 1], q'_{0100101}]$
 $[q_{0100101}, [1, 0], [1, 0], q'_{0100101}] [q'_{0100101}, [/ , /], [0, \downarrow], q''_{0100101}]$
 $[q''_{0100101}, [/ , 1], [0, 1], q'_{0100100}] [q'_{0100100}, [/ , /], [0, \downarrow], q''_{0100100}]$
 $[q'_{0100100}, [/ , 0], [0, 0], q'_{0100100}] [q'_{0100100}, [/ , /], [0, \downarrow], q'_{0100100}]$
 $[q'_{0100100}, [/ , 0], [0, 0], q'_{0100100}] [q'_{0100100}, [/ , /], [0, \downarrow], q'_{0100100}]$
 $[q'_{0100100}, [/ , 1], [0, 1], q'_0] [q'_0, [/ , /], [0, \downarrow], q_0]$

L (レフトシフト命令, [1010] で符号化) に対応する 4 項組:

$$\text{L } a_0 \longrightarrow \boxed{110110011011} \longrightarrow a_0$$

$[q_0, [/ , 0], [0, 0], q'_0] [q'_0, [/ , /], [0, \downarrow], q_0]$
 $[q_0, [/ , 1], [0, 1], q'_{01}] [q'_{01}, [/ , /], [0, \downarrow], q_{01}]$
 $[q_{01}, [/ , 0], [0, 0], q'_{010}] [q'_{010}, [/ , /], [0, \downarrow], q_{010}]$
 $[q_{010}, [/ , 1], [0, 1], q'_{0101}] [q'_{0101}, [/ , /], [0, \downarrow], q_{0101}]$
 $[q_{0101}, [/ , 0], [- , 0], q'_{01010}] [q'_{01010}, [/ , /], [0, \downarrow], q_{01010}]$
 $[q_{01010}, [/ , 0], [0, 0], q'_{010101}] [q'_{010101}, [/ , /], [0, \downarrow], q'_{010101}]$
 $[q'_{010101}, [/ , 1], [0, 1], q'_{010101}] [q'_{010101}, [/ , /], [0, \downarrow], q'_{010101}]$
 $[q'_{010101}, [/ , 0], [0, 0], q'_{010101}] [q'_{010101}, [/ , /], [0, \downarrow], q'_{010101}]$
 $[q'_{010101}, [/ , 1], [0, 1], q'_0] [q'_0, [/ , /], [0, \downarrow], q_0]$

R (ライトシフト命令, [1011] で符号化) に対応する 4 項組:

$$\text{R } a_0 \longrightarrow \boxed{10111111011} \longrightarrow a_0$$

$[q_0, [/ , 0], [0, 0], q'_0] [q'_0, [/ , /], [0, \downarrow], q_0]$
 $[q_0, [/ , 1], [0, 1], q'_{01}] [q'_{01}, [/ , /], [0, \downarrow], q_{01}]$
 $[q_{01}, [/ , 0], [0, 0], q'_{010}] [q'_{010}, [/ , /], [0, \downarrow], q_{010}]$
 $[q_{010}, [/ , 1], [0, 1], q'_{0101}] [q'_{0101}, [/ , /], [0, \downarrow], q_{0101}]$
 $[q_{0101}, [/ , 1], [+ , 1], q'_{01010}] [q'_{01010}, [/ , /], [0, \downarrow], q_{01010}]$
 $[q_{01010}, [/ , 1], [0, 1], q'_{010101}] [q'_{010101}, [/ , /], [0, \downarrow], q'_{010101}]$
 $[q'_{010101}, [/ , 1], [0, 1], q'_{010101}] [q'_{010101}, [/ , /], [0, \downarrow], q'_{010101}]$
 $[q'_{010101}, [/ , 0], [0, 0], q'_{010101}] [q'_{010101}, [/ , /], [0, \downarrow], q'_{010101}]$
 $[q'_{010101}, [/ , 1], [0, 1], q'_0] [q'_0, [/ , /], [0, \downarrow], q_0]$

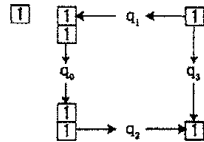
K (記号反転命令, [10011] で符号化) に対応する 4 項組:

$$\text{K } a_0 \longrightarrow \boxed{11001111110011} \longrightarrow a_0$$

$[q_0, [/ , 0], [0, 0], q'_0] [q'_0, [/ , /], [0, \downarrow], q_0]$
 $[q_0, [/ , 1], [0, 1], q'_{01}] [q'_{01}, [/ , /], [0, \downarrow], q_{01}]$
 $[q_{01}, [/ , 0], [0, 0], q'_{010}] [q'_{010}, [/ , /], [0, \downarrow], q_{010}]$
 $[q_{010}, [/ , 0], [0, 0], q'_{0100}] [q'_{0100}, [/ , /], [0, \downarrow], q_{0100}]$
 $[q_{0100}, [/ , 1], [0, 1], q'_{01001}] [q'_{01001}, [/ , /], [0, \downarrow], q_{01001}]$
 $[q_{01001}, [0, 1], [1, 1], q'_{010011}] [q'_{010011}, [/ , /], [0, \downarrow], q_{010011}]$
 $[q_{010011}, [1, 1], [0, 1], q'_{010011}]$
 $[q_{010011}, [/ , 1], [0, 1], q'_{010011}] [q'_{010011}, [/ , /], [0, \downarrow], q''_{010011}]$
 $[q''_{010011}, [/ , 1], [0, 1], q'_{0100100}] [q'_{0100100}, [/ , /], [0, \downarrow], q''_{0100100}]$
 $[q'_{0100100}, [/ , 0], [0, 0], q'_{0100100}] [q'_{0100100}, [/ , /], [0, \downarrow], q'_{0100100}]$
 $[q'_{0100100}, [/ , 0], [0, 0], q'_{0100100}] [q'_{0100100}, [/ , /], [0, \downarrow], q'_{0100100}]$
 $[q'_{0100100}, [/ , 1], [0, 1], q'_0] [q'_0, [/ , /], [0, \downarrow], q_0]$

方向転換 ([1] 又は [11] で符号化) に対応する 4 項組:

$[q_0, [/ , 0], [0, 0], q'_0] [q'_0, [/ , /], [0, \downarrow], q_0]$
 $[q_0, [/ , 1], [0, 1], q'_{01}] [q'_{01}, [/ , /], [0, \downarrow], q_{01}]$
 $[q_{01}, [/ , 1], [0, 1], q'_2] [q'_2, [/ , /], [0, \rightarrow], q_2]$
 $[q_2, [/ , 0], [0, 0], q'_2]$
 $[q_2, [/ , 1], [0, 1], q'_3] [q'_3, [/ , /], [0, \uparrow], q_3]$
 $[q_3, [/ , 0], [0, 0], q'_3]$
 $[q_3, [/ , 1], [0, 1], q'_1] [q'_1, [/ , /], [0, \leftarrow], q_1]$
 $[q_1, [/ , 0], [0, 0], q'_1]$
 $[q_1, [/ , 1], [0, 1], q'_{011}] [q'_{011}, [/ , /], [0, \downarrow], q'_{011}]$
 $[q'_{011}, [/ , 1], [0, 1], q'_0] [q'_0, [/ , /], [0, \downarrow], q_0]$



F (停止命令, [10000]で符号化) に対応する4項組:

$$F \quad q_0 \rightarrow [10101000]$$

- $[q_0, [/ , 0], [0, 0], q'_0] \quad [q'_0, [/ , /], [0, \downarrow], q_0]$
- $[q_0, [/ , 1], [0, 1], q'_{01}] \quad [q'_{01}, [/ , /], [0, \downarrow], q_{01}]$
- $[q_{01}, [/ , 0], [0, 0], q'_{010}] \quad [q'_{010}, [/ , /], [0, \downarrow], q_{010}]$
- $[q_{010}, [/ , 0], [0, 0], q'_{0100}] \quad [q'_{0100}, [/ , /], [0, \downarrow], q_{0100}]$
- $[q_{0100}, [/ , 0], [0, 0], q'_{01000}] \quad [q'_{01000}, [/ , /], [0, \downarrow], q_{01000}]$
- $[q_{01000}, [/ , 0], [0, 0], q_f]$

以上のように符号化したものを U_9 の2次元テープにそのまま埋め込む。図2は T_1 の場合の例である。

4 むすび

本稿では、9状態(2,8)記号2テープ2次元万能可逆チューリング機械を構築し、さらにそれを元にして、45状態(2,2)記号2テープ2次元万能可逆チューリング機械を構築した。

参考文献

- [1] C.H. Bennett: Logical reversibility of computation, IBM Journal of Reseach and Development, 17, 525-532 (1973).
- [2] M.L. Minsky: Computation: Finite and Infinite Machines, Prentice-Hall, Inc. (1967).
- [3] K. Morita, A. Shirasaki, Y. Gono: A 1-tape 2-symbol reversible turing machine, Trans. IEICE Japan, E72, 223-228 (1989).
- [4] Y. Rogozhin: Small universal turing machines, Theoretical Computer Science, 168, 215-240 (1996).

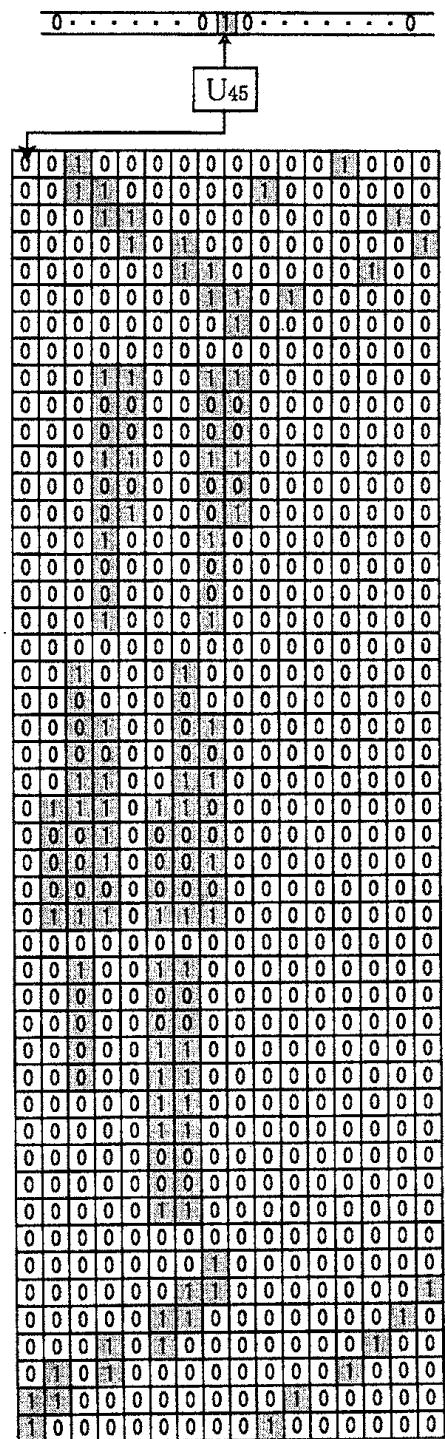


図2: T_1 を模倣する45状態(2,2)記号2テープ2次元万能可逆チューリング機械 U_{45}