Turing 機械の変種
Variants of Turing Machine

y.*

2018 年 5 月 20 日

最終更新日: 2018 年 5 月 23 日

概要

Turing 機械は人間の行う「計算」のモデル化によって得られた数学的概念である．Turing 機械の数学的な定式化は一通りではなく，様々なバリエーションが存在する．しかしながら，Turing 機械は多少の定義の変更によっては計算能力が変化しないという**頑健性** (robustness) を持つことが知られている．本稿では Turing 機械の定義に様々な制限・拡張を施し，それによって計算能力が変化しない様子を観察する．

Keywords: Turing 機械 (Turing machine), 多テープ Turing 機械 (multitape Turing machine), 非決定性 Turing 機械 (nondeterministic Turing machine), 頑健性 (robustness).

本稿で使用する Turing 機械の定義は「Turing 機械の定義と停止問題」[1] または Sipser の教科書 [3] を参照のこと．

1 Turing 機械の変種

1.1 ちょっとした変更

本小節ではウォーミングアップとして，Turing 機械の定義にいくつか簡単な変更を施し，変更後の機械がもとの Turing 機械と同じ計算能力を持つことを証明する．まずは，「計算能力が等しい」ということの意味を明確にしておこう．

定義 1.1 (計算モデルの等価性). 2 つの計算のモデル A と B が等価であるとは，任意の部分関数 $f\colon \Sigma^* \rightharpoonup \{\textsc{Yes}, \textsc{No}\}$ に対し，

$$f \text{ が A で計算可能} \iff f \text{ が B で計算可能}$$

となることである．

Turing 機械の元々の定義では，計算を 1 ステップ行うごとにヘッドを左右どちらかに動かさなければならなかった．ここではそれに加えて「その場にとどまる」ことも許すように Turing 機械の定義を変更してみよう．

定義 1.2 (静止可能 Turing 機械). 遷移関数の定義を $\delta\colon Q \times \Gamma \to Q \times \Gamma \times \{\mathrm{L}, \mathrm{R}\}$ から $\delta\colon Q \times \Gamma \to Q \times \Gamma \times \{\mathrm{L}, \mathrm{R}, \mathrm{S}\}$ に変更した機械をここでは**静止可能 Turing 機械** (Turing machine with stay put) と呼ぶ

* http://iso.2022.jp/

ことにする．

命題 1.3. 静止可能 Turing 機械は普通の Turing 機械と等価である．

証明. 計算モデルの等価性を証明するには，一方のモデルで計算可能な部分関数がもう一方のモデルでも計算可能であることを示せばよい．つまり今回の場合，任意の静止可能 Turing 機械に対し，同じ部分関数を計算する普通の Turing 機械が存在することを示し，また逆に任意の普通の Turing 機械に対して同じ部分関数を計算する静止可能 Turing 機械が存在することを示せばよい．簡単に言えば，お互いに相手をシミュレートできることを示せばよいのである．

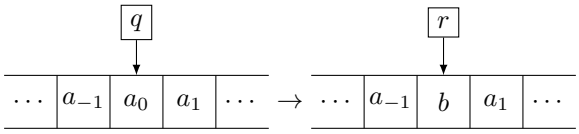
まず，任意の普通の Turing 機械 M をとる．このとき M は静止可能 Turing 機械でもあるから，M と同じ部分関数を計算する静止可能 Turing 機械として M そのものをとればよい．

次に，任意の静止可能 Turing 機械 $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$ をとる．M と等価な普通の Turing 機械 $M' = (Q', \Sigma', \Gamma, \delta', q_0, q_{\text{accept}}, q_{\text{reject}})$ を次のように構成する:

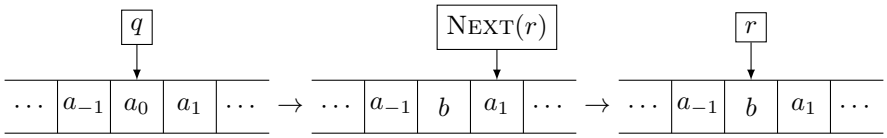
$$Q' := Q \cup \{\, \textsc{Next}(c) \mid c \in \Gamma \,\},$$

$$\delta'(q,a) := \begin{cases} (r, b, \mathrm{L}) & q \in Q, \delta(q,a) = (r,b,\mathrm{L}), \\ (r, b, \mathrm{R}) & q \in Q, \delta(q,a) = (r,b,\mathrm{R}), \\ (\textsc{Next}(r), b, \mathrm{R}) & q \in Q, \delta(q,a) = (r,b,\mathrm{S}), \\ (r, a, \mathrm{L}) & q = \textsc{Next}(r). \end{cases}$$

このように定めると，$\delta(q, a_0) = (r, b, \mathrm{S})$ というヘッドを動かさない M の動作



を，M' は



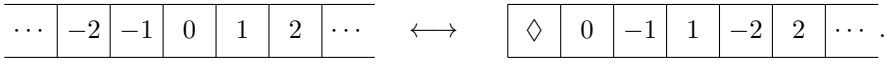
のようにして 2 ステップかけて再現する．ヘッドが左右に動く場合の動作は同じだから，M' は M と同じ部分関数を計算する． □

元々の Turing 機械の定義ではテープは右に無限に続いているが，左側には端があった．次はテープを両方向とも無限にしてみる．

定義 1.4. Turing 機械の定義において，テープが左側にも無限に長い機械を**両側無限テープ Turing 機械** (Turing machine with doubly infinite tape) という．

命題 1.5 ([3, 問題 3.11]). 両側無限テープ Turing 機械は普通の Turing 機械と等価である．

証明のアイデア. 両側無限テープに記録された内容を片側無限テープで再現するには次のようにする:



ここに ◊ はテープの左端を表すための記号である． □

証明. 任意に普通の Turing 機械 M をとる．M はそのまま両側無限テープ Turing 機械でもあるが，ひとつ注意しなければならないことは，普通の Turing 機械はヘッドが左端にある状態でさらにヘッドを左に動かそうとするとその場に留まるということである．だから M をそのまま両側無限テープ Turing 機械とみなして実行してしまうと，M を普通の Turing 機械として実行した場合と計算結果が異なるかもしれない．これを解決するには，M をあらかじめ「ヘッドがテープの左端を見ている状態でさらにヘッドを左に動かそうとすることはない」という条件を満たすように改造しておけばよい．詳細は「Post の対応問題」[2] を参照のこと．

次に，任意の両側無限テープ Turing 機械 $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\mathrm{accept}}, q_{\mathrm{reject}})$ をとる．このとき普通の Turing 機械 $M' = (Q', \Sigma, \Gamma', \delta', q_0', q_{\mathrm{accept}}, q_{\mathrm{reject}})$ を次のように構成する:

$$
\begin{aligned}
Q' &:= \{q_0', q_{\mathrm{accept}}, q_{\mathrm{reject}}\} \cup \{\, \textsc{Write}(c) \mid c \in \Gamma \,\} \\
&\quad \cup \{\textsc{Seek}, \textsc{Step}_2, \textsc{Step}_1, \textsc{GoToLeft}\} \\
&\quad \cup \bigcup_{q \in Q} \{q^+, q^-, \textsc{Left}(q^+), \textsc{Left}(q^-), \textsc{Right}(q^+), \textsc{Right}(q^-)\}, \\
\Gamma' &:= \Gamma \cup \{\Diamond\}, \\
\delta'(q_0', a) &:= \begin{cases} (\textsc{Write}(a), \Diamond, \mathrm{R}) & (a \neq \text{␣}), \\ (q_0^+, \Diamond, \mathrm{R}) & (a = \text{␣}), \end{cases} \\
\delta'(\textsc{Write}(c), a) &:= \begin{cases} (\textsc{Write}(a), c, \mathrm{R}) & (a \neq \text{␣}), \\ (\textsc{Seek}, c, \mathrm{L}) & (a = \text{␣}), \end{cases} \\
\delta'(\textsc{Seek}, a) &:= \begin{cases} (\textsc{Seek}, a, \mathrm{L}) & (a \notin \{\Diamond, \text{␣}\}), \\ (\textsc{Step}_2, a, \mathrm{R}) & (a \in \{\Diamond, \text{␣}\}), \end{cases} \\
\delta'(\textsc{Step}_2, a) &:= (\textsc{Step}_1, a, \mathrm{R}), \\
\delta'(\textsc{Step}_1, a) &:= \begin{cases} (\textsc{Write}(a), \text{␣}, \mathrm{R}) & (a \neq \text{␣}), \\ (\textsc{GoToLeft}, \text{␣}, \mathrm{L}) & (a = \text{␣}), \end{cases} \\
\delta'(\textsc{GoToLeft}, a) &:= \begin{cases} (\textsc{GoToLeft}, a, \mathrm{L}) & (a \neq \Diamond), \\ (q_0^+, \Diamond, \mathrm{R}) & (a = \Diamond), \end{cases} \\
\delta'(q^+, a) &:= \begin{cases} (\textsc{Right}(r^+), b, \mathrm{R}) & (\delta(q, a) = (r, b, \mathrm{R}), q \in Q, a \in \Gamma), \\ (\textsc{Left}(r^+), b, \mathrm{L}) & (\delta(q, a) = (r, b, \mathrm{L}), q \in Q, a \in \Gamma), \end{cases} \\
\delta'(q^-, a) &:= \begin{cases} (\textsc{Left}(r^-), b, \mathrm{L}) & (\delta(q, a) = (r, b, \mathrm{R}), q \in Q, a \in \Gamma), \\ (\textsc{Right}(r^-), b, \mathrm{R}) & (\delta(q, a) = (r, b, \mathrm{L}), q \in Q, a \in \Gamma), \\ (q^+, \Diamond, \mathrm{R}) & (a = \Diamond), \end{cases} \\
\delta'(\textsc{Right}(q^\pm), a) &:= (q^\pm, a, \mathrm{R}), \\
\delta'(\textsc{Left}(q^-), a) &:= (q^-, a, \mathrm{L}), \\
\delta'(\textsc{Left}(q^+), a) &:= \begin{cases} (q^+, a, \mathrm{L}) & (a \neq \Diamond), \\ (\textsc{Right}(q^-), \Diamond, \mathrm{R}) & (a = \Diamond), \end{cases} \\
\delta'(q_{\mathrm{accept}}^\pm, a) &:= (q_{\mathrm{accept}}, a, \mathrm{R}), \\
\delta'(q_{\mathrm{reject}}^\pm, a) &:= (q_{\mathrm{reject}}, a, \mathrm{R}).
\end{aligned}
$$

M' はまず左端のマスに $\Diamond$ を書き込み，入力文字列の全体を 1 マスぶんだけ右にずらす．その後入力文字列の全体を少しずつ右へずらすことで，各文字の間に空白文字␣を挿入する．状態の肩に付いている符号 $\pm$ は，q^+ なら両側無限テープの右側にいることを，q^- なら左側にいることを表している．この M' は M と同じ部分

関数を計算する．□

次に，ヘッドを左へ 1 マス動かすことはできないが，代わりにヘッドをテープの左端まで一気に動かせるような機械を考える．

定義 1.6. 通常の Turing 機械の定義において，遷移関数を $\delta\colon Q \times \Gamma \to Q \times \Gamma \times \{\mathrm{R}, \textsc{Reset}\}$ に変更した機械を**左リセット Turing 機械** (Turing machine with left reset) という．ここで，$\delta(q, a) = (r, b, \textsc{Reset})$ は状態 q で文字 a を読んでいるとき，テープに b を書き，状態を r に変更し，機械のヘッドをテープの左端にジャンプさせることを意味する．

命題 1.7 ([3, 問題 3.12]). 左リセット Turing 機械は普通の Turing 機械と等価である．

証明のアイデア. 左リセット Turing 機械でもヘッドを 1 マスだけ左へ動かす動作を再現できることを示せばよい．読者は次のように考えるかもしれない: 「左へ 1 マスだけ移動するには，一旦テープの左端へ移動し，それからもといたマスのひとつ左のマスまで右に移動し続ければよい．そのためには，今までに見たマスの全てに (ヘンゼルとグレーテルがやったように) 印を付けておいて，ひとたび左リセットされたら，印の付いているマスのうち最も右のマスまで移動すればよい.」ところが，この作戦は**うまくいかない**．なぜなら，ヘッドが現在見ているマスが「印の付いているマスのうち最も右のマス」かどうかは，**さらに右に 1 マス移動してみなければわからない**からである．実はうまくやればこの問題は回避することができるのだが,[*1] ここでは次のもっと簡単な方法を用いる: ヘッドを左に動かすのではなく，テープの内容を右へずらせばよいのである．

普通の Turing 機械においてヘッドを 1 マスだけ左へ動かす動作を左リセット Turing 機械で再現するためのアイデアを図 1 に示す．□

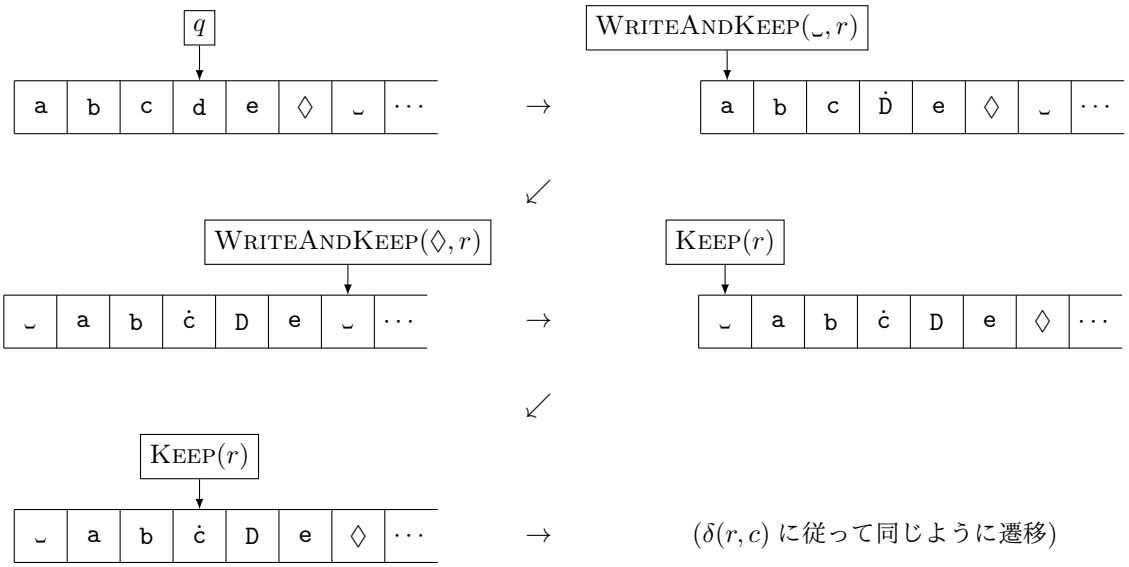


図 1: $\delta(q, \mathtt{d}) = (r, \mathtt{D}, \mathrm{L})$ を左リセット Turing 機械で再現する

[*1] 普通の Turing 機械のテープの内容を $a_1 \cdots a_n$ とするとき，左リセット Turing 機械で $a_1 \cdots a_n \Diamond a_1 \cdots a_n$ のようにテープの内容の同一のコピーを 2 つ保持しておくとうまくいく．しかしこの方法は正確に書き下そうとするとかなり煩雑になるのでここでは用いない．

証明. 任意に左リセット Turing 機械 M をとる．このとき，M と等価な通常の Turing 機械 M' を容易に構成することができる．(あらかじめテープの左端に目印となる記号 $\Diamond$ を書き込んでおき，テープの左端へ行きたいときは $\Diamond$ を目指して左へ移動し続ければよい．)

次に，任意に普通の Turing 機械 $M = (Q, \Sigma, \Gamma, \delta, q'_0, q_{\mathrm{accept}}, q_{\mathrm{reject}})$ をとる．例によって M は「ヘッドがテープの左端を見ている状態でさらにヘッドを左に動かそうとすることはない」としてよい．このとき左リセット Turing 機械 $M' = (Q', \Sigma, \Gamma', \delta', q'_0, q_{\mathrm{accept}}, q_{\mathrm{reject}})$ を次のように構成する:

$$\begin{aligned} Q' &:= Q \cup \{q'_0\} \cup \{\, \textsc{Keep}(q) \mid q \in Q \setminus \{q_{\mathrm{accept}}, q_{\mathrm{reject}}\} \,\} \\ &\quad \cup \{\, \textsc{WriteAndKeep}(a, q) \mid a \in \Gamma \cup \{\Diamond\}, q \in Q \setminus \{q_{\mathrm{accept}}, q_{\mathrm{reject}}\} \,\}, \\ \Gamma' &:= \Gamma \cup \{\, \dot{a} \mid a \in \Gamma \,\} \cup \{\Diamond\}, \end{aligned}$$

$q \in Q \setminus \{q_{\mathrm{accept}}, q_{\mathrm{reject}}\}, a, b \in \Gamma$ として

$$\delta'(q'_0, a) := \begin{cases} (q'_0, a, \mathrm{R}) & (a \neq \textvisiblespace), \\ (q_0, \Diamond, \textsc{Reset}) & (a = \textvisiblespace), \end{cases}$$

$$\delta'(q, a) := \begin{cases} (r, b, \mathrm{R}) & (\delta(q, a) = (r, b, \mathrm{R})), \\ (\textsc{WriteAndKeep}(\textvisiblespace, r), \dot{b}, \textsc{Reset}) & (\delta(q, a) = (r, b, \mathrm{L}), b \notin \{q_{\mathrm{accept}}, q_{\mathrm{reject}}\}), \\ (r, b, \mathrm{L}) & (\delta(q, a) = (r, b, \mathrm{L}), b \in \{q_{\mathrm{accept}}, q_{\mathrm{reject}}\}), \end{cases}$$

$$\begin{aligned} \delta'(q, \dot{a}) &:= \delta'(q, a), \\ \delta'(q, \Diamond) &:= (\textsc{WriteAndKeep}(\Diamond, q), \dot{\textvisiblespace}, \mathrm{R}), \\ \delta'(\textsc{Keep}(q), a) &:= (\textsc{Keep}(q), a, \mathrm{R}) \quad (a \in \Gamma), \\ \delta'(\textsc{Keep}(q), \dot{a}) &:= \delta'(q, a), \\ \delta'(\textsc{WriteAndKeep}(a, q), b) &:= (\textsc{WriteAndKeep}(b, q), a, \mathrm{R}), \\ \delta'(\textsc{WriteAndKeep}(a, q), \dot{b}) &:= (\textsc{WriteAndKeep}(b, q), \dot{a}, \mathrm{R}), \\ \delta'(\textsc{WriteAndKeep}(\Diamond, q), b) &:= (\textsc{Keep}(q), \Diamond, \textsc{Reset}). \end{aligned}$$

M' は右端のマスに $\Diamond$ を書き込み，その後は上で示したアイデアの通りに動作する．よってこの M' が M と同じ部分関数を計算する． □

最後に，テープのマスを書き換えられる回数を制限した機械を考える．

定義 1.8. テープのそれぞれのマス (入力部分も含める) について，書かれた内容を高々 2 回だけ変更できる機械を **2 回書き込み Turing 機械** (write-twice Turing machine) という．3 回目の変更を試みた時点で計算結果は未定義 ($M(w)\uparrow$) とする．**1 回書き込み Turing 機械** (write-once Turing machine) も同様に定義される．

驚くべきことに，1 回書き込み Turing 機械は普通の Turing 機械と等価であることが示せる．これを示すために，まずは 2 回書き込み Turing 機械の場合を証明する．

補題 1.9. 2 回書き込み Turing 機械は普通の Turing 機械と等価である．

証明の概略. 2 回書き込み Turing 機械を普通の Turing 機械でシミュレートできることはよい．(書き換えた回数を各マスにメモしておけばよい．)

次に，普通の Turing 機械を 2 回書き込み Turing 機械でシミュレートする方法を与える．テープの内容を

書き換えられる回数が制限されているので，テープの内容を直接書き換えるのではなく，遷移後のテープの内容をテープの右側へ追記していく．

例えば，$\delta(q, \mathtt{c}) = (r, \mathtt{C}, \mathrm{L})$ という 1 ステップの動作を次のようにシミュレートする (テープの内容の変化のみ記す):

$\cdots$	$\Diamond$	a	b	$\dot{\mathtt{c}}$	d	$\Diamond$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\cdots$
$\cdots$	$\Diamond$	$\check{\mathtt{a}}$	b	$\dot{\mathtt{c}}$	d	$\Diamond$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\cdots$
$\cdots$	$\Diamond$	$\check{\mathtt{a}}$	b	$\dot{\mathtt{c}}$	d	$\Diamond$	a	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\cdots$
$\cdots$	$\Diamond$	$\check{\mathtt{a}}$	$\check{\mathtt{b}}$	$\dot{\mathtt{c}}$	d	$\Diamond$	a	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\cdots$
$\cdots$	$\Diamond$	$\check{\mathtt{a}}$	$\check{\mathtt{b}}$	$\dot{\mathtt{c}}$	d	$\Diamond$	a	$\dot{\mathtt{b}}$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\cdots$
$\cdots$	$\Diamond$	$\check{\mathtt{a}}$	$\check{\mathtt{b}}$	$\check{\mathtt{c}}$	d	$\Diamond$	a	$\dot{\mathtt{b}}$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\cdots$
$\cdots$	$\Diamond$	$\check{\mathtt{a}}$	$\check{\mathtt{b}}$	$\check{\mathtt{c}}$	d	$\Diamond$	a	$\dot{\mathtt{b}}$	C	$\sqcup$	$\sqcup$	$\sqcup$	$\cdots$
$\cdots$	$\Diamond$	$\check{\mathtt{a}}$	$\check{\mathtt{b}}$	$\check{\mathtt{c}}$	$\check{\mathtt{d}}$	$\Diamond$	a	$\dot{\mathtt{b}}$	C	$\sqcup$	$\sqcup$	$\sqcup$	$\cdots$
$\cdots$	$\Diamond$	$\check{\mathtt{a}}$	$\check{\mathtt{b}}$	$\check{\mathtt{c}}$	$\check{\mathtt{d}}$	$\Diamond$	a	$\dot{\mathtt{b}}$	C	d	$\sqcup$	$\sqcup$	$\cdots$
$\cdots$	$\Diamond$	$\check{\mathtt{a}}$	$\check{\mathtt{b}}$	$\check{\mathtt{c}}$	$\check{\mathtt{d}}$	$\Diamond$	a	$\dot{\mathtt{b}}$	C	d	$\sqcup$	$\Diamond$	$\cdots$

ヘッドの位置を $\dot{}$ で表し，コピー済みであることの印として $\check{}$ を用いている．各々のマスは，最初に文字を書き込むために 1 回，コピー済みの印を付けるためにもう 1 回変更される． □

命題 1.10 ([3, 問題 3.10]). 1 回書き込み Turing 機械は普通の Turing 機械と等価である．

証明の概略. 補題 1.9 より，2 回書き込み Turing 機械を 1 回書き込み Turing 機械でシミュレートできることを言えばよい．これを実現するためには，2 回書き込み Turing 機械における 1 マスを，1 回書き込み Turing 機械の 2 マスで表せばよい．例えば，普通の Turing 機械における $\delta(q, \mathtt{a}) = (r, \mathtt{A}, \mathrm{R})$ という 1 ステップの動作を次のようにシミュレートする:

$\cdots$	$\Diamond$	$\Diamond$	$\dot{\mathtt{a}}$	$\sqcup$	b	$\sqcup$	$\Diamond$	$\Diamond$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\cdots$
$\cdots$	$\Diamond$	$\Diamond$	$\dot{\mathtt{a}}$	$\check{}$	b	$\sqcup$	$\Diamond$	$\Diamond$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\cdots$
$\cdots$	$\Diamond$	$\Diamond$	$\dot{\mathtt{a}}$	$\check{}$	b	$\sqcup$	$\Diamond$	$\Diamond$	A	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\cdots$
$\cdots$	$\Diamond$	$\Diamond$	$\dot{\mathtt{a}}$	$\check{}$	b	$\check{}$	$\Diamond$	$\Diamond$	A	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\cdots$
$\cdots$	$\Diamond$	$\Diamond$	$\dot{\mathtt{a}}$	$\check{}$	b	$\check{}$	$\Diamond$	$\Diamond$	A	$\sqcup$	$\dot{\mathtt{b}}$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\cdots$
$\cdots$	$\Diamond$	$\Diamond$	$\dot{\mathtt{a}}$	$\check{}$	b	$\check{}$	$\Diamond$	$\Diamond$	A	$\sqcup$	$\dot{\mathtt{b}}$	$\sqcup$	$\sqcup$	$\Diamond$	$\Diamond$	$\cdots$

以上のようにして，書かれている文字を保持するマスとコピー済みの印を付けるマスを分けることで，各マスが高々 1 回しか変更されないようにできる．ただし，計算開始時だけは入力文字列が 1 マスおきに書かれていないので，まず 1 つおきになるように内容をコピーするところから始める． □

1.2 多テープ Turing 機械

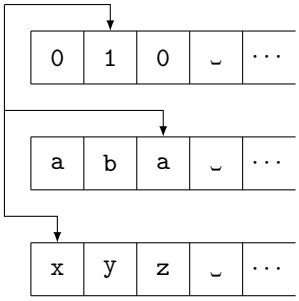
本小節では Turing 機械のテープの本数を増やすことを考える．これまでと同様に，実はテープの本数をいくら増やしても計算能力は変わらない．

定義 1.11 (多テープ Turing 機械 (multitape Turing machine)). k を正の整数とする．**k テープ Turing 機械** (k-tape Turing machine) とは，k 本のテープと k 個の独立したヘッドを持つ Turing 機械である．k テープ Turing 機械の遷移関数は $\delta\colon Q \times \Gamma^k \to Q \times \Gamma^k \times \{\mathrm{L}, \mathrm{R}\}^k$ という形をしている．k テープ Turing 機械への入力は 1 本目のテープにだけ書き込まれる．

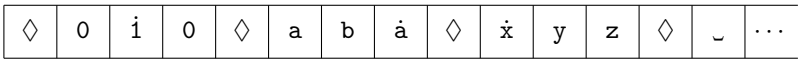
命題 1.12 ([3, 定理 3.13]). 任意の k に対し，k テープ Turing 機械は普通の Turing 機械と等価である．

証明の概略. k テープ Turing 機械の動作を模倣する普通の Turing 機械を構成すればよい．

これを達成する方法は少なくとも 2 つある．一つは，k 本のテープの内容を直列に並べておき，これまでと同様にヘッドの位置に印を付けておく方法である．例えば $k = 3$ の場合，k テープ Turing 機械が

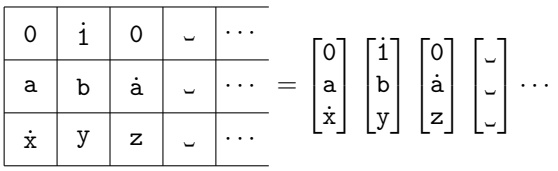


という状況であれば，対応する普通の Turing 機械のテープは



となる．

もう一つの方法は，テープアルファベットの “縦ベクトル” を 1 つの文字とみなす方法である．この方法では，上の状況は



というベクトルの列で表される．こちらのやり方の方がヘッドの移動量が少なくて済むので経済的である．[*2] □

[*2] しかしながら，この方法でも 2 つのヘッドが逆方向へ移動するような場合にはシミュレーションに時間がかかってしまう ($O(f(n))$ 時間 k テープ Turing 機械をシミュレートするのに $O(f(n)^2)$ 時間かかる)．実は，ヘッドの位置を表す印 ˙ の方を動かすのではなく，テープの内容の方を動かすことで $O(f(n))$ 時間 k テープ Turing 機械と等価な $O(f(n)\log f(n))$ 時間 2 テープ Turing 機械 (1 テープではない) を構成できることが知られている [5].

1.3 非決定性 Turing 機械

定義 1.13. **非決定性 Turing 機械** (nondeterministic Turing machine) M は，遷移関数が $\delta\colon Q \times \Gamma \to \mathcal{P}(Q \times \Gamma \times \{\mathrm{L}, \mathrm{R}\})$ という形をした機械である (ここで $\mathcal{P}$ は冪集合を表す)．計算の各ステップにおいて，$\delta(q, a) = \{(q_1, a_1, D_1), (q_2, a_2, D_2), \ldots, (q_m, a_m, D_m)\}$ だったとすると，M は m 個に “分裂” し，m 通りの全ての遷移の可能性を並列に実行する．受理するような実行の経路が少なくとも 1 つ存在するとき，M は入力を受理する (停止しない実行の経路があってもよい)．全ての実行の経路で停止して拒否されるとき，M は入力を拒否する ($\delta(q, a) = \emptyset$ となる場合も拒否されたとみなす)．それ以外のときの計算結果は未定義とする．[*3]

例 1.14. 非決定性 Turing 機械 $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\mathrm{accept}}, q_{\mathrm{reject}})$ を次のように定義する:

$$
\begin{aligned}
Q &:= \{q_0, q_1, q_2, q_{\mathrm{accept}}, q_{\mathrm{reject}}\},\\
\Sigma &:= \{\mathtt{a}, \mathtt{b}\},\\
\Gamma &:= \{\mathtt{a}, \mathtt{b}, \mathtt{␣}\},
\end{aligned}
$$

δ	a	b	␣
q_0	$\{(q_0, \mathtt{a}, \mathrm{R}), (q_1, \mathtt{a}, \mathrm{R})\}$	$\{(q_0, \mathtt{b}, \mathrm{R})\}$	$\{(q_{\mathrm{reject}}, \mathtt{␣}, \mathrm{R})\}$
q_1	$\{(q_{\mathrm{reject}}, \mathtt{a}, \mathrm{R})\}$	$\{(q_2, \mathtt{b}, \mathrm{R})\}$	$\{(q_{\mathrm{reject}}, \mathtt{␣}, \mathrm{R})\}$
q_2	$\{(q_{\mathrm{reject}}, \mathtt{a}, \mathrm{R})\}$	$\{(q_{\mathrm{reject}}, \mathtt{b}, \mathrm{R})\}$	$\{(q_{\mathrm{accept}}, \mathtt{␣}, \mathrm{R})\}$.

この M は「ab で終わる文字列」を受理する機械であり，例えば入力 abab に対しては図 2 のように動作する．

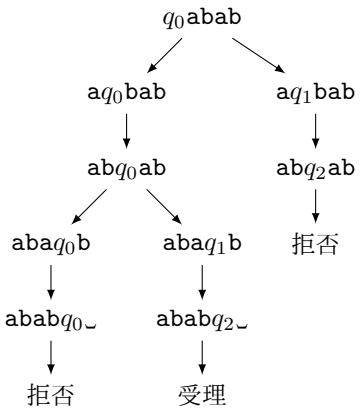


図 2: 非決定性 Turing 機械の動作

命題 1.15 ([3, 定理 3.16]). 非決定 Turing 機械は普通の Turing 機械と等価である．

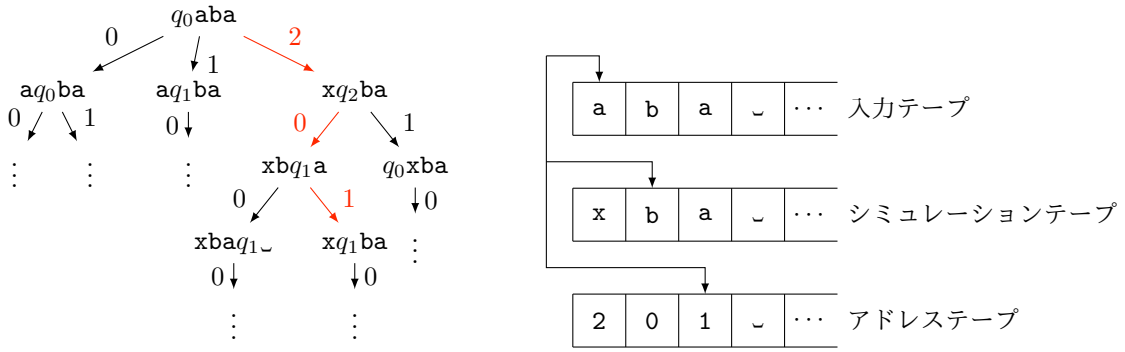
証明の概略. 命題 1.12 より，任意の非決定性 Turing 機械 N を 3 テープ Turing 機械 M でシミュレートできることを示せば十分である．N の計算は図 3a のように木構造をなす．あらかじめ $Q \times \Gamma \times \{\mathrm{L}, \mathrm{R}\}$ の元に番号を付けて大小を比較できるようにしておくことで，木の各々の節点 (計算状況) に一意的な数列を割り当てることができる (これをアドレスと呼ぶ).

[*3] 停止しない実行の経路が一つでもあれば結果を未定義とする流儀もある．

M の 3 つのテープはそれぞれ入力テープ，シミュレーションテープ，アドレステープと呼ばれる．入力テープは N への入力を保持しておくためのテープで，内容は書き換えずに読み込み専用のテープとして用いる．シミュレーションテープは計算をアドレスに沿って再現するためのテープである．アドレステープはアドレス (シミュレートすべき実行の経路) を保持しておくためのテープである．M のシミュレーションは次のように行われる．

1. シミュレーションテープの内容を全て消去し，入力テープの内容をシミュレーションテープにコピーする．
2. 現在のアドレステープの内容に従ってシミュレーションを実行する．
3. シミュレーション中に受理状態に入ったら受理する．
4. 現在の深さの全ての実行の経路で計算が停止し，しかも全ての経路で拒否されたならば拒否する．
5. アドレスを 1 つカウントアップし，最初に戻る．

図 3a の木においてアドレスが 201 のときの M の計算状況を図 3b に示す．[*4] □



(a) N の計算　　(b) M によるシミュレーション

図 3: 非決定性 Turing 機械 N の 3 テープ Turing 機械 M によるシミュレーション

2　自然数値関数を計算する Turing 機械

ここまでは与えられた文字列に対し Yes/No を返す Turing 機械を考えてきた．本節では与えられた自然数の組 $(x_1, \ldots, x_k)$ に対し自然数の値 $f(x_1, \ldots, x_k)$ を返す Turing 機械を考える．すなわち，部分関数 $f\colon \mathbb{N}^k \rightharpoonup \mathbb{N}$ を計算する Turing 機械を考える．

定義 2.1.　自然数値関数を計算する Turing 機械は，通常の Turing 機械の定義に以下の変更を施したものである．[*5]

- $q_{\text{accept}}, q_{\text{reject}}$ の代わりにただひとつの停止状態 q_{halt} を用意する．

[*4] 「アドレステープを用意したり毎回テープの内容を消去するのは無駄なのではないか」と思う読者もいるかもしれない．しかし，「図 3a の木を計算が停止するまで下り，拒否されたら直前の分岐点まで木を上に戻る」という方法 (深さ優先探索) では，受理される経路と計算が停止しない経路が混在している場合にシミュレーションを正しく行うことができない．そのため，図 3a の木は「同じ深さの節点を全てたどってから，次の深さへ行く」という方法 (幅優先探索) でたどられる必要がある．

[*5] この定義は Soare [4] を参考にした．

- 入力アルファベットを $\Sigma = \{\mathtt{1}\}$ とする．
- 関数の引数が $(x_1, \ldots, x_k) \in \mathbb{N}^k$ であるときは，テープの左端から連続する $x_i + 1$ 個の $\mathtt{1}$ を空白記号␣で区切って並べ，残りを空白記号␣で埋めた状態 $\mathtt{1}^{x_1+1}␣\mathtt{1}^{x_2+1}␣\cdots␣\mathtt{1}^{x_k+1}␣␣\cdots$ から計算を始める．
- q_{halt} に到達した時点でテープ上にある空白記号以外の文字の個数 (有限個) を出力値とする．
- q_{halt} に到達しなければ計算結果は未定義とする．

例えば，$q_0 = q_{\mathrm{halt}}$ という機械 (テープの内容を全く変更せず最初のステップで停止する機械) を 1 変数関数 $f\colon \mathbb{N} \rightharpoonup \mathbb{N}$ を計算する機械と考えると，$f(x) = x + 1$ である．

2.1 二進アルファベット Turing 機械

定義 2.1 において，テープアルファベットを $\Gamma = \{\mathtt{1}, ␣\}$ に制限した機械をここでは**二進アルファベット Turing 機械** (Turing machine with binary alphabet) と呼ぶことにする．

命題 2.2. 二進アルファベット Turing 機械は普通の Turing 機械と等価である．

証明のアイデア. ␣と $\mathtt{1}$ だけを用いて，普通の Turing 機械のテープアルファベット Γ の各元を 2 進列として表現すればよい．例えば $|\Gamma| \leq 16$ であれば，Γ の元を表すには 4 ビットあれば十分である．よってこのとき $\Gamma = \{␣, \mathtt{1}, \mathtt{a}, \mathtt{b}, \mathtt{c}, \ldots\}$ とすると

$$
\begin{aligned}
␣ &\leftrightarrow ␣␣␣␣\\
\mathtt{1} &\leftrightarrow \mathtt{1}␣␣␣\\
\mathtt{a} &\leftrightarrow ␣\mathtt{1}␣␣\\
\mathtt{b} &\leftrightarrow \mathtt{11}␣␣\\
\mathtt{c} &\leftrightarrow ␣␣\mathtt{1}␣\\
&\vdots
\end{aligned}
$$

と表される．したがって，テープの 4 マスを一つのかたまりとしてシミュレーションを実行すればよい． □

証明. 任意に普通の Turing 機械 $M = (Q, \Gamma, \delta, q_0, q_{\mathrm{halt}})$ をとる．簡単のため，$|\Gamma| \leq 16$ で 1 変数関数を計算する場合を考える．一般の場合も同様である．初めに，M は「計算の停止時，␣と $\mathtt{1}$ 以外の記号がテープ上に残ることはない」と仮定してよい．なぜなら，M に「計算を終了するときに，テープ上の ␣, $\mathtt{1}$ 以外の記号を全て $\mathtt{1}$ に書き換える」という動作を追加しておけばよいからである．

$\Gamma' := \{␣, \mathtt{1}\}$ とおく．ビット列 $s \in \Gamma'^4$ に対応する Γ の文字を $\mathsf{chr}(s)$ と書くことにする．このとき二進アルファベット Turing 機械 $M' = (Q', \Gamma', \delta', q'_0, q'_{\mathrm{halt}})$ を次のように構成する:

$$
\begin{aligned}
Q'_{\mathrm{preprocess}} &:= \{q'_0, q'_1\} \cup \{\,\textsc{Scanning}(s) \mid s \in \Gamma' \cup \Gamma'^2\,\}\\
&\quad \cup \{\,\textsc{Write}(s) \mid s \in \Gamma' \cup \Gamma'^2 \cup \Gamma'^3\,\} \cup \{\,\textsc{Continue}(n) \mid n = 0, 1\,\}\\
&\quad \cup \{\,\textsc{GoToLeft}(c) \mid c \in \Gamma'\,\},\\
Q'_{\mathrm{main}} &:= \{\,\textsc{Scanning}(q, s) \mid q \in Q, s \in \Gamma' \cup \Gamma'^2 \cup \Gamma'^3\,\}\\
&\quad \cup \{\,\textsc{Write}(q, s, D) \mid q \in Q, s \in \Gamma' \cup \Gamma'^2 \cup \Gamma'^3, D \in \{\mathrm{L}, \mathrm{R}\}\,\}\\
&\quad \cup \bigcup_{\substack{q \in Q\\ n=0,1,2}} \{\textsc{MoveLeft}(q, n), \textsc{MoveRight}(q, n)\} \cup \{\textsc{GoToNextScan}\},\\
Q' &:= Q'_{\mathrm{preprocess}} \cup Q'_{\mathrm{main}},
\end{aligned}
$$

$c, c_0, c_1, c_2, c_3 \in \Gamma', q \in Q$ として

$$
\begin{aligned}
\delta'(q'_0, \mathtt{1}) &:= (q'_1, \mathtt{1}, \mathrm{R}),\\
\delta'(q'_1, \text{␣}) &:= (\textsc{GoToLeft}(\text{␣}), \text{␣}, \mathrm{L}),\\
\delta'(q'_1, \mathtt{1}) &:= (\textsc{Scanning}(\mathtt{1}), \text{␣}, \mathrm{R}),\\
\delta'(\textsc{Scannig}(c_0), c_1) &:= (\textsc{Scanning}(c_0, c_1), \text{␣}, \mathrm{R}),\\
\delta'(\textsc{Scannig}(c_0, c_1), c_2) &:= (\textsc{Write}(c_0, c_1, c_2), \text{␣}, \mathrm{R}),\\
\delta'(\textsc{Write}(c_0, c_1, c_2), \mathtt{1}) &:= (\textsc{Write}(c_0, c_1, c_2), \mathrm{R}),\\
\delta'(\textsc{Write}(c_0, c_1, c_2), \text{␣}) &:= (\textsc{Write}(c_1, c_2), c_0, \mathrm{R}),\\
\delta'(\textsc{Write}(c_1, c_2), \text{␣}) &:= (\textsc{Write}(c_2), c_1, \mathrm{R}),\\
\delta'(\textsc{Write}(c_2), \text{␣}) &:= (\textsc{Continue}(0), c_2, \mathrm{L}),\\
\delta'(\textsc{Continue}(0), c) &:= (\textsc{Continue}(1), c, \mathrm{L}),\\
\delta'(\textsc{Continue}(1), \mathtt{1}) &:= (\textsc{Continue}(1), \mathtt{1}, \mathrm{L}),\\
\delta'(\textsc{Continue}(1), \text{␣}) &:= (q'_0, \text{␣}, \mathrm{R}),\\
\delta'(\textsc{GoToLeft}(\text{␣}), c) &:= (\textsc{GoToLeft}(c), c, \mathrm{L}),\\
\delta'(\textsc{GoToLeft}(\mathtt{1}), \text{␣}) &:= (\textsc{GoToLeft}(\text{␣}), \text{␣}, \mathrm{L}),\\
\delta'(\textsc{GoToLeft}(\mathtt{1}), \mathtt{1}) &:= (\textsc{Scanning}(q_0, \mathtt{1}), \mathtt{1}, \mathrm{R}),\\
\delta'(\textsc{Scanning}(q, c_0), c_1) &:= (\textsc{Scanning}(q, c_0, c_1), c_1, \mathrm{R}),\\
\delta'(\textsc{Scanning}(q, c_0, c_1), c_2) &:= (\textsc{Scanning}(q, c_0, c_1, c_2), c_2, \mathrm{R}),\\
\delta'(\textsc{Scanning}(q, c_0, c_1, c_2), c_3) &:= (\textsc{Write}(r, d_0, d_1, d_2, D), d_3, \mathrm{L})\\
&\qquad (\delta(q, \mathsf{chr}(c_0, c_1, c_2, c_3)) = (r, \mathsf{chr}(d_0, d_1, d_2, d_3), D)),\\
\delta'(\textsc{Write}(q, c_0, c_1, c_2, D), c) &:= (\textsc{Write}(r, c_0, c_1, D), c_2, \mathrm{L}),\\
\delta'(\textsc{Write}(q, c_0, c_1, D), c) &:= (\textsc{Write}(r, c_0, D), c_1, \mathrm{L}),\\
\delta'(\textsc{Write}(q, c_0, \mathrm{L}), c) &:= (\textsc{MoveLeft}(q, 2), c_0, \mathrm{L}),\\
\delta'(\textsc{Write}(q, c_0, \mathrm{R}), c) &:= (\textsc{MoveRight}(q, 2), c_0, \mathrm{R}),\\
\delta'(\textsc{MoveLeft}(q, 2), c) &:= (\textsc{MoveLeft}(q, 1), c, \mathrm{L}),\\
\delta'(\textsc{MoveLeft}(q, 1), c) &:= (\textsc{MoveLeft}(q, 0), c, \mathrm{L}),\\
\delta'(\textsc{MoveLeft}(q, 0), c) &:= (\textsc{GoToNextScan}(q), c, \mathrm{L}),\\
\delta'(\textsc{MoveRight}(q, 2), c) &:= (\textsc{MoveRight}(q, 1), c, \mathrm{R}),\\
\delta'(\textsc{MoveRight}(q, 1), c) &:= (\textsc{MoveRight}(q, 0), c, \mathrm{R}),\\
\delta'(\textsc{MoveRight}(q, 0), c) &:= (\textsc{GoToNextScan}(q), c, \mathrm{R}),\\
\delta'(\textsc{GoToNextScan}(q), c) &:= \begin{cases} (\textsc{Scanning}(q, c), c, \mathrm{R}) & (q \neq q_{\mathrm{halt}}),\\ (q'_{\mathrm{halt}}, c, \mathrm{R}) & (q = q_{\mathrm{halt}}). \end{cases}
\end{aligned}
$$

M' への入力文字列が仮に $\mathtt{11111}\text{␣}\cdots$ だったとしよう．M' はまず $Q'_{\mathrm{preprocess}}$ の状態を用いて入力文字列を 3 文字ずつ末尾へ移動させる．

$$
\begin{aligned}
& \mathtt{11111}\text{␣}\cdots\\
\to\ & \mathtt{1}\text{␣␣␣}\mathtt{1111}\text{␣}\cdots\\
\to\ & \mathtt{1}\text{␣␣␣}\mathtt{1}\text{␣␣␣}\mathtt{111}\text{␣}\cdots\\
\to\ & \mathtt{1}\text{␣␣␣}\mathtt{1}\text{␣␣␣}\mathtt{1}\text{␣␣␣}\mathtt{11}\text{␣}\cdots\\
\to\ & \mathtt{1}\text{␣␣␣}\mathtt{1}\text{␣␣␣}\mathtt{1}\text{␣␣␣}\mathtt{1}\text{␣␣␣}\mathtt{1}\text{␣}\cdots.
\end{aligned}
$$

その後テープの左端へ移動してからシミュレーションを開始するのだが，このときテープの左端に到達したことを検出するために「テープの左端にいるときにさらに左へ移動しようとしたときはその場に留まる」という性質を利用する．テープ上に連続する 1 がなくなったため，2 回連続で 1 を見たらテープの左端にいると判断できるのである．

最後に，M の $\delta(q, \mathsf{chr}(\text{␣}, 1, \text{␣}, \text{␣})) = (r, \mathsf{chr}(1, \text{␣}, \text{␣}, 1), \mathrm{R})$ という動作が M' によってどのようにシミュレートされるかを図 4 に示しておく． □

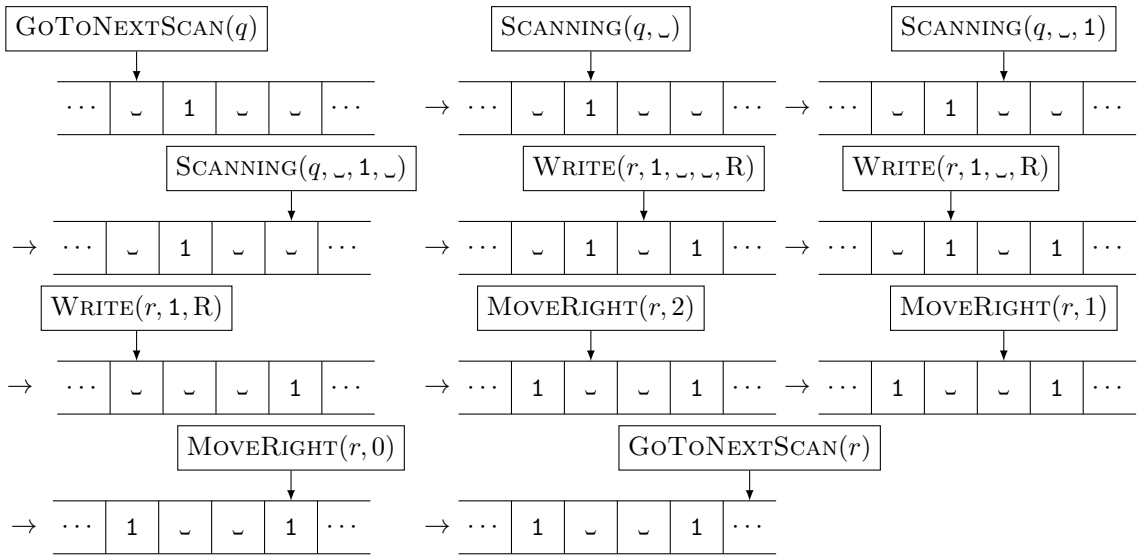


図 4: 二進アルファベット Turing 機械 M' で普通の Turing 機械 M をシミュレートする

参考文献

[1] y., Turing 機械の定義と停止問題 (2018), http://iso.2022.jp/math/undecidable-problems/files/turing-machine-and-the-halting-problem.pdf.

[2] y., Post の対応問題 (2018), http://iso.2022.jp/math/undecidable-problems/files/post-correspondence-problem.pdf.

[3] M. Sipser (太田和夫・田中圭介 監訳, 阿部正幸・植田広樹・藤岡淳・渡辺治 訳), 計算理論の基礎 [原著第 2 版] 2. 計算可能性の理論, 共立出版, 2008.

[4] R. I. Soare, *Turing Computability: Theory and Applications*, Springer, 2016.

[5] F. C. Hennie, R. E. Stearns, Two-Tape Simulation of Multitape Turing Machines, *J. ACM* **13** no. 4 (1966) 533–546, https://doi.org/10.1145/321356.321362.

変更履歴

2018/05/20　公開
2018/05/23　微修正