計算理論

Thomas Zeugmann

北海道大学 大学院情報科学研究科
アルゴリズム研究室

http://www-alg.ist.hokudai.ac.jp/∼thomas/ToC/

第 12 回：チューリング機械
(日本語訳：湊 真一，大久保 好章)

チューリング機械 I

部分帰納的関数 (partial recursive function) を扱った後は，アラン-チューリングにより提案されたチューリング機械 (Turing machine) に注目することにしよう．

彼のアイデアは，第 11 回の講義で述べた，アルゴリズムの 4 つの性質を用いて「直感的な計算可能」な関数という概念を形式的に表すことである．まず始めに彼が行ったのは，機械の基本的な演算を，すべてのアルゴリズムを実行可能な最小限のレベルに絞り込むことであった．話を単純にするため，ここでは 1 テープチューリング機械 (one-tape Turing machine) を考える．

チューリング機械 II

1 テープ チューリング機械は，多数のセルに分割された無限長のテープを持つ．個々のセルは，ただ 1 つのテープ記号 (tape-symbol) を記入できる．入力値が実際に書き込まれているセル以外には，初期値として記号 $*$ が入っていると仮定する．我々は，テープのセルを図 1 のように数えるものとする．

	−5	−4	−3	−2	−1	0	1	2	3	4	5
	*	*	*	*	*	b_1	b_2	b_3	*	*	

図 1: 入力 $b_1b_2b_3$ を持つチューリング機械のテープ.

チューリング機械 III

さらに，チューリング機械は読み書きヘッドを持つ．このヘッドは 1 度に 1 つのセルを見ることができる．そしてこの機械は，有限個の状態を持ち，実行する命令の集合を持っている．この機械は，最初は必ず開始状態 z_s にあり，ヘッドは入力の最も左のセル 0 を見ている．我々はヘッドの位置を矢印で表すこととする．

チューリング機械 III

さらに，チューリング機械は読み書きヘッドを持つ．このヘッドは 1 度に 1 つのセルを見ることができる．そしてこの機械は，有限個の状態を持ち，実行する命令の集合を持っている．この機械は，最初は必ず開始状態 z_s にあり，ヘッドは入力の最も左のセル 0 を見ている．我々はヘッドの位置を矢印で表すこととする．

さて，この機械は以下のように動作する．ある状態 z において，テープ記号 b を読んだとき，そのセルに記号 b' を書き込み，状態を z' に遷移し，ヘッドを左 (L と書く) または右 (R と書く) に移動するか，ヘッドを動かさない (N と書く)，という動作を，$(z, \mathrm{b}, \mathrm{b}', \mathrm{H}, z')$ の組 (ただし $\mathrm{H} \in \{\mathrm{L}, \mathrm{N}, \mathrm{R}\}$) で与えられるチューリング機械の命令にしたがって実行する．1 つの命令の実行を**ステップ (*step*)** と呼ぶ．

チューリング機械 IV

この機械は，特別 (*distinguished*) な状態 z_f (終了状態) に到達したとき，停止するものとする．以上より，チューリング機械を以下のように定義することができる．

定義 1

$M = [B, Z, A]$ は，以下を満たすとき**決定性 *1* テープ チューリング機械 (*deterministic one-tape Turing machine*)** と呼ばれる．すなわち，B, Z, A が空でない有限集合であって，$B \cap Z = \emptyset$ かつ

(1) $\mathrm{card}(B) \geqslant 2$ $\quad (B = \{*, |, \ldots\})$ $\quad$ (テープ記号),

(2) $\mathrm{card}(Z) \geqslant 2$ $\quad (Z = \{z_s, z_f, \ldots\})$ $\quad$ (状態集合),

(3) $A \subseteq Z \setminus \{z_f\} \times B \times B \times \{L, N, R\} \times Z$ $\quad$ (命令集合), ただしそれぞれの $z \in Z \setminus \{z_f\}$ と，それぞれの $b \in B$ に対して，ただ *1* つの *5* 個組 $(z, b, \cdot, \cdot, \cdot)$ が決まっていること．

チューリング機械 V

命令集合 A は，しばしば表形式で書かれる．例えば，

	$*$	$\vert$	b_2	$\dots$	b_m
z_s	$b'Nz_3$				
z_1	·				
·	·				
·	·				
·	·				
z_n	·				

チューリング表

もしも命令集合が小さければ，(z, b, b', H, z') と書く代わりに，$zb \;\to\; b'Hz'$(ただし $H \in \{L, N, R\}$) と書く方が便利なことが多い．また，チューリング機械 M の命令集合のことを，通常，M の**プログラム (*program*)** と言う．

チューリング計算 I

次に，チューリング機械がどのようにして関数を計算するかを説明しなければならない．我々がまず興味を持つ対象は，$\mathbb{N}^n$ から $\mathbb{N}$ への関数，すなわち，$f\colon \mathbb{N}^n \to \mathbb{N}$ である．したがって入力は，$(x_1, \ldots, x_n) \in \mathbb{N}^n$ なる組である．

ここでは，x_i と x_{i+1} を分けるため，特殊なテープ記号 # を予約することにする．さらに，以下では話を簡単にするため，数値は 1 進数 (unary) で符号化されているものとする．例えば 0 は $*$ と表現され，1 は | と表し，2 は ||，3 は ||| とし，以下同様とする．なお，チューリング計算の計算量 (*complexity*) を気にしない限りは，この記法は何ら制約にはならない．

チューリング計算 II

さらに，次の記法を導入すると便利である．任意の関数 $f\colon \mathbb{N}^n \to \mathbb{N}$ において，ある入力 $(x_1, \ldots, x_n) \in \mathbb{N}^n$ の組に対する $f(x_1, \ldots, x_n)$ の値が未定義であるとき，$f(x_1, \ldots, x_n) \uparrow$ と書くことにする．一方， $f(x_1, \ldots, x_n)$ が定義されている場合は，$f(x_1, \ldots, x_n) \downarrow$ と書くこととする．

チューリング計算 III

定義 2

任意のチューリング機械 M と，任意の $n \in \mathbb{N}^+$ と，任意の関数 $f\colon \mathbb{N}^n \to \mathbb{N}$ を考える．もしもすべての $(x_1,\ldots,x_n) \in \mathbb{N}^n$ に対して，以下の条件を満たすならば，M は関数 f を**計算する *(compute)***と言う．

(1) もしも $f(x_1,\ldots,x_n)\downarrow$ であって，$x_1\#\ldots\#x_n$ が M の空のテープに書かれていて，M の初期ヘッド位置が $x_1\#\ldots\#x_n$ の最も左にあり，初期状態が z_s であるならば，M は有限ステップ実行後に状態 Z_f で停止する．さらに，もしも $f(x_1,\ldots,x_n) = 0$ であれば，M が状態 z_f において読める記号は $*$ である．もしも $f(x_1,\ldots,x_n) \neq 0$ であれば，M が状態 z_f において読めるセルから始まる連続した $|$ の列（左から右へ向かって読むとする）が計算結果を表す (図 2 を参照).

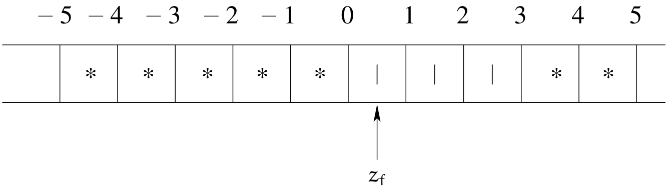

図 2: 計算結果 3 (|||) を示すチューリング機械のテープ.

定義 3 (続き)

(2) もしも $f(x_1, \ldots, x_n) \uparrow$ であって，$x_1\#\ldots\#x_n$ が M の空のテープに (セル 0 から) 書かれていて，M の初期ヘッド位置が $x_1\#\ldots\#x_n$ の最も左にあり，初期状態が z_s であるならば，M は停止しない.

チューリング計算 IV

f^n_M という記法により，チューリング機械 M により計算される $\mathbb{N}^n$ から $\mathbb{N}$ への関数を表すことにする．

定義 4

任意の $n \in \mathbb{N}^+$ と任意の関数 $f\colon \mathbb{N}^n \to \mathbb{N}$ において，$f^n_M = f$ となるようなチューリング機械 M が存在するならば，関数 f は**チューリング計算可能 (*Turing computable*)** であると言う．さらに，以下のように定義する．

$$
\begin{aligned}
\mathcal{T}^n &= \text{すべての } n \text{ 入力のチューリング計算可能な関数の集合.}\\
\mathcal{T} &= \bigcup_{n \geqslant 1} \mathcal{T}^n = \text{すべてのチューリング計算可能な関数の集合.}
\end{aligned}
$$

チューリング計算 V

ここで，どのような関数がチューリング計算可能であるかという自然な疑問が生ずる．その答えは次の定理により与えられる．

定理 1

チューリング計算可能な関数のクラスは，部分帰納的関数のクラスと等しい．すなわち，$\mathcal{T} = \mathcal{P}$ である．

証明 I

$\mathcal{P} \subseteq \mathcal{T}$ を示すには，以下を証明すれば十分である．

(1) 関数 Z, S, V がチューリング計算可能である．

(2) チューリング計算可能な関数のクラスが，第 11 回の講義で定義された演算 (2.1) から (2.6) に関して閉じている．

証明 II

定数 0 関数を計算するチューリング機械は，次のようにして容易に定義できる． $M = [\{*, |\}, \{z_s, z_f\}, A]$ とする．ただし A は以下のような命令集合である．

$$
\begin{aligned}
z_s| &\quad\longrightarrow\quad |Lz_f \\
z_s* &\quad\longrightarrow\quad *Nz_f\ .
\end{aligned}
$$

つまり，もしも入力がゼロでなければ，M はヘッドを 1 つ左へ動かして停止する．我々のチューリング機械の定義では，M は-1 番地のセルの $*$ を見ることになり, その出力は 0 となる．もしも入力がゼロであった場合は，M は 0 番地のセルの $*$ を見ているので，そのまま書き換えず，ヘッドを動かさずに停止する．明らかにすべての $x \in \mathbb{N}$ に対して $f^1_M(x) = 0$ である．

証明 III

後者関数 (successor function) S は次のようなチューリング機械 $M = [\{*, |\}, \{z_s, z_f\}, A]$ により計算できる．ただし A は以下のような命令集合である．

$$
\begin{aligned}
z_s| &\quad\rightarrow\quad |Lz_s \\
z_s* &\quad\rightarrow\quad |Nz_f\ .
\end{aligned}
$$

つまり，この機械は入力に 1 個の $|$ を書き足すだけで停止する．したがって，すべての $x \in \mathbb{N}$ に対して $f_M^1(x) = S(x)$ となる．

証明 IV

さらに，前者関数 (predecessor function) $M = [\{*, |\}, \{z_s, z_f\}, A]$ により計算できる．ただし A は以下のような命令集合である．

$$\begin{array}{lcl} z_s| & \longrightarrow & *Rz_f \\ z_s* & \longrightarrow & *Nz_f\ . \end{array}$$

このチューリング機械は，0 番地のセルの $|$ を読んで，これを消去してヘッドを 1 つ右へ動かす，あるいは $*$ を読んでヘッドを動かさずに停止するか，いずれかとなる．したがって，すべての $x \in \mathbb{N}$ に対して $f_M^1(x) = V(x)$ となる．以上で (1) の部分については証明できた．

証明 V

次に (2) の部分の証明の概略を示す．これは一連の主張により示される．

主張 1. (架空変数の導入)
$n \in \mathbb{N}^+$ とする．もしも $\tau \in \mathcal{T}^n$ かつ
$\psi(x_1, \ldots, x_n, x_{n+1}) = \tau(x_1, \ldots, x_n)$ ならば，$\psi \in \mathcal{T}^{n+1}$ である．

主張 1 が成り立つことは直感的に明らかである．ψ を計算するために必要なのは，x_{n+1} を入力テープから消去し，その後，ヘッドを初期位置に戻し，τ のためのチューリングプログラムを開始することである．したがって，$\psi \in \mathcal{T}^{n+1}$ となる．詳細は省略する．

証明 VI

主張 2. (変数の同一視)
$n \in \mathbb{N}^+$ とする．もしも $\tau \in \mathcal{T}^{n+1}$ かつ
$\psi(x_1, \dots, x_n) = \tau(x_1, \dots, x_n, x_n)$ ならば，$\psi \in \mathcal{T}^n$ である．

主張 2 を証明するためには，最後の変数（すなわち x_n）をコピーするチューリングプログラムがあれば良い．よって，最初のテープの内容

$$* \ * \ x_1\#\dots\#x_n \ \ * \ *$$

を

$$* \ * \ x_1\#\dots\#x_n\#x_n * \ *$$

に書き換え，その後，ヘッドを初期位置に戻し，M を，τ を計算するためのプログラムの初期状態にセットする．そして τ のプログラムを開始すればよい．したがって $\psi \in \mathcal{T}^n$ である．ここでも詳細は省略する．

証明 VII

主張 3. (変数の置換)
$n \in \mathbb{N}^+$, $n \geqslant 2$ とし，$i \in \{1, \ldots, n\}$ とする．
もしも $\tau \in \mathcal{T}^n$ かつ
$\psi(x_1, \ldots, x_i, x_{i+1}, \ldots, x_n) = \tau(x_1, \ldots, x_{i+1}, x_i, \ldots, x_n)$ ならば，
$\psi \in \mathcal{T}^n$ である．

主張 3 は，主張 2 と同様に (*mutatis mutandis;* 必要な変更を加えて) 示すことができる．したがって，ここでは証明は省略する．

証明 VIII

主張 4. (合成)
$n \in \mathbb{N}$ および $m \in \mathbb{N}^+$ とする．さらに $\tau \in \mathcal{T}^{n+1}, \psi \in \mathcal{T}^m$ とし，$\phi(x_1, \ldots, x_n, y_1, \ldots, y_m) = \tau(x_1, \ldots, x_n, \psi(y_1, \ldots, y_m))$ とする．このとき $\phi \in \mathcal{T}^{n+m}$ である．

主張 4 の証明は少しややこしい．明らかなアイデアとしては，y_1 の最初の記号までヘッドを右に移動し，ψ を計算するチューリングプログラムを開始すれば良さそうである．もしも $\psi(y_1, \ldots, y_m) \uparrow$ ならば，ϕ を計算する機械もまた，入力 $x_1, \ldots, x_n, y_1, \ldots, y_m$ に対して発散する．しかし，もしも $\psi(y_1, \ldots, y_m) \downarrow$ ならば，我々のゴールは，新しいテープ内容

$$* \ * \ x_1\#\ldots\#x_n\#\psi(y_1, \ldots, y_m) * \ *$$

を作って，x_1 の最初の記号を指すようにヘッドを左に戻すことである．これで，τ のチューリングプログラムを開始すれば良い．

証明 IX

そこで我々が克服すべき問題は，$\psi(y_1, \ldots, y_m)$ の計算によって，x_i が上書きされないことを保証する，ということである．
よって，以下の補題が必要である．

補題 2 (M^+)

すべてのチューリング機械 M に対して，以下のようなチューリング機械 M^+ が存在する．

(1) すべての n に対して $f_M^n = f_{M^+}^n$ である．

(2) M^+ は計算開始時の初期セルの位置よりも左には移動しない．

(3) すべての $x_1, \ldots, x_n$ に対して，もしも $f_{M^+}^n(x_1, \ldots, x_n) \downarrow$ ならば，初期ヘッド位置と同じ場所で計算が停止し，計算結果よりも右側のテープには $*$ しか書かれていない．

補題 M^+ の証明

補題 M^+ を示すためのアイデアは以下の通りである．M^+ は以下のように動作する．

- すべての入力を 1 セルずつ右へ移動させる．
- 初期セルに特別な記号をマークする．これを L と呼ぶことにしよう．
- 移動した入力の右側にある最初のセルに特別な記号をマークする．これを E と呼ぶことにしよう．
- 次に示す 3 つの例外を除き，M と同様の動作をする．

補題 M^+ の証明

- もしも M^+ が E とマークされたセルに到達したときには（このときの状態を z と呼ぶことにする），マーク E を 1 セルだけ右に移動し，その後ヘッドを 1 セル左に戻し，そのセル（元は E だったセル）に $*$ を書き込む．そしてその位置（元は E だったが今は $*$ になったセル）から M の状態 z でのシミュレーションを続行する．
- もしも M が状態 z において，L と書かれたセルに来てしまったときには，L と E の間にあるすべてのテープの内容（E を含むが L は除く）を，1 つずつ右へ移動させる．L の右隣のセルに $*$ を書き込み，M^+ は，このセルでの M のシミュレーションを続行する．

補題 M^+ の証明

- もしも M が停止したときには，M^+ は，すべての計算結果を，その最初の記号が L のセルの位置に来るまで，左に移動させる．さらに，移動した計算結果のすぐ右側から E までのすべてのセルに $*$ を書き込み，その後，計算結果の最も左側のセルにヘッドを戻す．そして M^+ もまた停止する．

以上で補題 M^+ が証明された．
補題 M^+ を持って，主張 4 が上記の通り示された．

残りの原始再帰および μ-再帰に関する主張は，演習問題とする．
以上で $\mathcal{P} \subseteq \mathcal{T}$ が示された．

証明 X

最後に $\mathcal{T} \subseteq \mathcal{P}$ を示す必要がある．$n \in \mathbb{N}+$ および $f \in \mathcal{T}^n$ とし，f を計算する任意のチューリング機械を M とする．関数 t (time; 時間) と r (result; 結果) を以下のように定義する．

$$t(x_1,\ldots,x_n,y) = \begin{cases} 1\,, & M \text{ が } x_1,\ldots,x_n, \text{ に対して} \\ & \quad \text{高々} y \text{ ステップ実行後に停止するとき;} \\ 0\,, & \text{それ以外のとき.} \end{cases}$$

$$r(x_1,\ldots,x_n,y) = \begin{cases} f(x_1,\ldots,x_n)\,, & t(x_1,\ldots,x_n,y) = 1 \text{ のとき;} \\ 0\,, & \text{それ以外のとき.} \end{cases}$$

これで，t, $r \in \mathit{Prim}$ を示すことができる．さらに，クリーニは，上で定義した t と r に関する次の標準形定理 (normal form theorem) を示した．

証明 XI

定理 3 (Kleene)

すべての $f \in \mathcal{T}^n$, $n \in \mathbb{N}^+$ に対して，関数 t, $r \in \mathit{Prim}$ が存在し，すべての $x_1, \dots, x_n \in \mathbb{N}^n$ で以下を満たす．

$$f(x_1, \dots, x_n) = r(x_1, \dots, x_n, \mu y[t(x_1, \dots, x_n, y) = 1]) \tag{1}$$

この定理の証明は本講義の範囲を越えるので，ここでは示さない．

t と r の原始帰納性と，等式 (1) によって，$\mathcal{T} \subseteq \mathcal{P}$ が成り立つ．なぜならば，この等式によって，μ-演算を 1 回だけ適用すれば (演算 (2.6))，計算結果の関数 r は演算 (2.4) により求められるからである．以上で $f \in \mathcal{P}$ が示された． ∎

チャーチの提唱

上記の定理は，基礎的な認識論に関する重要性を持っている．我々は全く異なる方向から議論を始めたにも関わらず，最終的に同じ計算可能関数にたどり着いた．つまり,「直感的に計算可能」な関数を形式的に表すための，異なるアプローチが提案されたということである．これらのアプローチには他にも，ポストのアルゴリズム，マルコフのアルゴリズム，ランダムアクセス機械 (Random Access machine; RAM)，チャーチの λ-計算がある．これらの形式化手法はすべて，同じ計算可能関数の集合を定義しているということが，後にわかった．つまり結果として得られる関数のクラスは，チューリング計算可能な関数と等しい．この事実が，チャーチの有名な提唱を導いた．

チャーチの提唱 (Church's Thesis). 「直感的に計算可能」な関数のクラスは，チューリング計算可能な関数のクラスと等しい．

万能チューリング機械 I

ここで我々は，ただ 1 つのチューリング機械が存在して，それによってすべての部分帰納的関数が計算できることを示そう．

万能チューリング機械 I

ここで我々は，ただ 1 つのチューリング機械が存在して，それによってすべての部分帰納的関数が計算できることを示そう．

まず，対関数 (pairing function) を利用した我々の結果を用いれば，任意の n 個組の自然数を 1 つの自然数で符号化できることが容易にわかる．しかもこの符号化は，すでに述べたように，原始帰納的である．そこで以下では，$\mathbb{N}$ から N へのすべての部分帰納的関数の集合を $\mathcal{P}$ と表記することとする．

万能チューリング機械 I

ここで我々は，ただ 1 つのチューリング機械が存在して，それによってすべての部分帰納的関数が計算できることを示そう．

まず，対関数 (pairing function) を利用した我々の結果を用いれば，任意の n 個組の自然数を 1 つの自然数で符号化できることが容易にわかる．しかもこの符号化は，すでに述べたように，原始帰納的である．そこで以下では，$\mathbb{N}$ から N へのすべての部分帰納的関数の集合を $\mathcal{P}$ と表記することとする．

次に，任意の部分帰納的関数 $\psi(i, x)$ を考える．すなわち，$\psi\colon \mathbb{N}^2 \to \mathbb{N}$ である．ここで，もしも第 1 引数 i を固定すれば，1 変数の部分帰納的関数 ψ_i が得られる．

万能チューリング機械 II

すると, ψ で計算できるすべての 1 変数関数は次のように図示できる (図 3 参照).

$$
\begin{array}{lllllll}
\psi_0 & \psi(0,0) & \psi(0,1) & \psi(0,2) & \psi(0,3) & \psi(0,4) & \ldots \\
\psi_1 & \psi(1,0) & \psi(1,1) & \psi(1,2) & \psi(1,3) & \psi(1,4) & \ldots \\
\psi_2 & \psi(2,0) & \psi(2,1) & \psi(2,2) & \psi(2,3) & \psi(2,4) & \ldots \\
\psi_3 & \psi(3,0) & \psi(3,1) & \psi(3,2) & \psi(3,3) & \psi(3,4) & \ldots \\
\psi_4 & \psi(4,0) & \psi(4,1) & \psi(4,2) & \psi(4,3) & \psi(4,4) & \ldots \\
\psi_5 & \psi(5,0) & \ldots & & & & \\
\cdot & & & & & & \\
\cdot & & & & & & \\
\cdot & & & & & & \\
\psi_i & \ldots & \ldots & & & & \\
\ldots & & & & & &
\end{array}
$$

図 3: すべての ψ_i を表す 2 次元配列表.

万能チューリング機械 III

例をあげると，$\psi(i, x) = ix$ を考えるならば，例えば $\psi_7(x) = 7x$ となる．したがって，すべての関数 $\psi \in \mathcal{P}^2$ は，一種の番号付け *(numbering)* であると言うことが正当化できる．

定義 5

以下が成り立つとき，関数 $\psi \in \mathcal{P}^2$ は，$\mathcal{P}$ に対して**万能** ***(universal)*** であると言う．

$$\{\psi_i \mid i \in \mathbb{N}\} = \mathcal{P} .$$

万能チューリング機械 III

例をあげると，$\psi(i, x) = ix$ を考えるならば，例えば $\psi_7(x) = 7x$ となる．したがって，すべての関数 $\psi \in \mathcal{P}^2$ は，一種の番号付け *(numbering)* であると言うことが正当化できる．

定義 5

以下が成り立つとき，関数 $\psi \in \mathcal{P}^2$ は，$\mathcal{P}$ に対して万能 ***(universal)*** であると言う．

$$\{\psi_i \mid i \in \mathbb{N}\} = \mathcal{P}\,.$$

ここで明らかに興味深い疑問は $\mathcal{P}$ に対して万能な $\psi \in \mathcal{P}^2$ が存在するかどうかである．もしも $\mathcal{P}$ に対して万能な ψ が存在するならば，$\mathcal{P} = \mathcal{T}$ であるから，ψ もまたチューリング計算可能である．したがって，ψ を計算する任意のチューリング機械を万能チューリング機械 ***(universal Turing machine)*** として解釈できることになる．次の定理で，万能な ψ が存在することが確かとなる．

万能チューリング機械 IV

定理 4

$\mathcal{P}$ に対して万能な関数 $\psi \in \mathcal{P}^2$ が存在する.

万能チューリング機械 V

このアイデアは容易に説明できる．我々の定理 $\mathcal{P} = \mathcal{T}$ より，すべての $\tau \in \mathcal{P}$ に対して，$f_M^1 = \tau$ となるようなチューリング機械 M が存在する．そこで，すべてのチューリング機械を自然数で符号化することを目指そう．それには，$cod(M) \in \mathbb{N}$ となるような単射の一般帰納的関数 *cod* が必要である．さらに，このアイデアが成立するためには，次のような一般帰納的関数 *decod* も必要である．

$$decod(cod(M)) = (M \text{ のプログラム}).$$

もしも *decod* への入力 i が，チューリング機械の正しい符号でない場合には，$decod(i) = 0$ と定義する．

万能チューリング機械 VI

こうして，万能関数 ψ は，2 つの入力変数 i と x を持つチューリング機械 $\mathcal{U}$ により記述できる．動作開始時には，通常と同様，まず $decod(i)$ を計算する．もしも $decod(i) = 0$ ならば，関数 Z(定数 0) を計算する．そうでなければ，$decod(i)$ が返した機械 M のプログラムをシミュレートする．

万能チューリング機械 VI

こうして，万能関数 ψ は，2 つの入力変数 i と x を持つチューリング機械 $\mathcal{U}$ により記述できる．動作開始時には，通常と同様，まず $decod(\mathrm{i})$ を計算する．もしも $decod(\mathrm{i}) = 0$ ならば，関数 Z(定数 0) を計算する．そうでなければ，$decod(\mathrm{i})$ が返した機械 M のプログラムをシミュレートする．

この動作を実現するには，以下の追加条件を満たさなければならない．

(1) $\mathcal{U}$ は $decod(\mathrm{i})$ によって得られたプログラムを上書きしてはならない．

(2) $\mathcal{U}$ は，有限個のテープ記号と有限の状態集合で実現しなければならない．

万能チューリング機械 VII

次に，どのようにしてすべての条件が実現できるかを手短に述べる．読みやすくするため，以下では，テープ記号を b_i と書き，状態集合を $z_s, z_f, \ldots$ と始めることにする．そして，

$$M = [\{b_1, \ldots, b_m\}, \{z_s, z_f, z_1, \ldots, z_k\}, A]$$

が与えられるとする．

万能チューリング機械 VIII

ここで我々は，以下のような符号を用いる．
(0^n は，ちょうど n 個のゼロからなる文字列を表す．)

$$
\begin{aligned}
cod(\mathrm{L}) &= 101 \\
cod(\mathrm{R}) &= 1001 \\
cod(\mathrm{N}) &= 10001 \\
cod(z_s) &= 10^4 1 \\
cod(z_f) &= 10^6 1 \\
cod(z_\ell) &= 10^{2(\ell+3)}1 && \text{for all } \ell \in \{1,\ldots,k\}, \\
cod(b_\ell) &= 10^{2(\ell+1)+1}1 && \text{for all } \ell \in \{1,\ldots,m\}.
\end{aligned}
$$

万能チューリング機械 IX

そして命令集合の符号化は，各部分の符号を連結することによって行われる．すなわち，

$$cod(zb \;\to\; b'Hz') = cod(z)cod(b)cod(b')cod(H)cod(z') \,.$$

例えば，$z_s b_1 \;\to\; b_2 N z_1$ は次のように符号化される．

$$10^4 110^5 110^7 11000110^8 1 \,.$$

ここで，m 個のテープ記号と $k+2$ の状態があるので，全部で $m(k+1)$ 種類の命令が存在し，ある特定の順序づけを仮定すると $I_1, \ldots, I_{m(k+1)}$ と書き表すことができる．最終的に，M のプログラムは，そのすべての命令の符号を連結することにより，次のように符号化できる．

$$cod(M) = cod(I_1) \cdots cod(I_{m(k+1)}) \,.$$

この文字列は自然数を 2 進数で表したものと解釈できる．

万能チューリング機械 X

cod が単射であることは容易に分かる．すなわち，
もし $M \neq M'$ ならば $cod(M) \neq cod(M')$ である．

さらに，もし以上に示したような *cod* を使うならば，*decode* が，許容される文字列を与えられたかどうかを判定する手間が省ける．つまり，文字列から M のプログラムを直接読み取ることができる．

最後に，シミュレーションがどのように行われるかを述べなければならない．まず，U が M のプログラムを破壊しないことを保証しなくてはならない．これは本質的には，補題 M^+ で述べたやり方とほぼ同じである．

万能チューリング機械 XI

後は，条件 (2) がどのように実現されるかの説明が残っている．明らかに，U はシミュレーションの途中で，自分自身の状態集合を使って M の動作状態を記憶することはできない．なぜならば，無限個の状態が必要になる可能性があるからである，しかし，U は M の動作状態を，テープにマークすることができる．(例えば強調文字を使うなど)．

U が有限個のテープ記号しか使わないということを保証するために，U は，M のテープアルファベットの中の記号 b_ℓ を直接使わずに，$cod(b_\ell) = 10^{2(\ell+1)+1}1$ だけを用いることにする．これなら，シミュレーションを行うために，たった 2 つのテープ記号しか必要としない． 以下，詳細は省略する．
このようにして，チューリング機械 U は，定理 $\mathcal{P} = \mathcal{T}$ を使えば，部分帰納的関数 $\psi \in \mathcal{P}^2$ によって表現可能である．

言語の受理 I

次に，チューリング機械が言語 L を受理するということが，何を意味するかを定義する．

定義 6

もしすべての文字列 $w \in \Sigma^*$ に対して次の条件を満たすとき，言語 $L \subseteq \Sigma^*$ はチューリング機械 M で受理される *(accepted)* と言う．もし w が，M の空テープに (0 番地のセルから) 書かれていて，チューリング機械 M が w の最も左の記号から，状態 z_s で実行開始するとき，M は有限ステップの実行後に状態 z_f で停止する．さらに，

(1) もし $w \in L$ ならば，M が状態 z_f で見ているセルには $|$ が入っている．この場合を $M(w) = |$ と書くこともある．

(2) もし $w \notin L$ ならば，M が状態 z_f で見ているセルには $*$ が入っている．この場合を $M(w) = *$ と書くこともある．

言語の受理 II

もちろん，言語 $L \subseteq \Sigma^*$ がチューリング機械 $M = [B, Z, A]$ に受理されるためには，常に $\Sigma \subseteq B$ を仮定しなければならない．
さらに，すべてのチューリング機械 M に対して

$$L(M) =_{df} \{w \mid w \in \Sigma^* \ \wedge \ M(w) = |\}$$

を定義し，M により受理される言語を $L(M)$ と表す．

テキストにいくつか例があるので見ておくこと．

Thank you!