計算可能性理論特論・講義ノート *†

木原 貴行
名古屋大学 情報学部・情報学研究科

最終更新日: 2018 年 2 月 1 日

目次

* 本講義ノートは，2017 年度秋 1 期および秋 2 期開講の名古屋大学大学院情報学研究科における講義「計算可能性理論特論 1」および「計算可能性理論特論 2」の内容をまとめたものである．
† 講義のページ： http://www.math.mi.i.nagoya-u.ac.jp/~kihara/teach.html

1 計算モデルと計算理論

1.1 チューリング機械

文字列を書くことのできるテープと，その上の文字列の読み書きをするためのヘッドを持つ機械 TM を思い浮かべよう．この機械 TM は，有限種類の状態を取ることができる．

この機械 TM の現在の状況として，テープに $\mathtt{a}_0\mathtt{a}_1\mathtt{a}_2\ldots\mathtt{a}_n$ という文字列が書かれており，ヘッドは $\mathtt{a}_k$ の位置にあり，現在の状態は q であるとしよう．機械 TM の絵を毎回書くのは面倒なので，この状況を以下によって表すこととする．

$$[\ \sqcup\mathtt{a}_0\mathtt{a}_1\mathtt{a}_2\ldots\mathtt{a}_{k-1}\ \boxed{q}\Rrightarrow\mathtt{a}_k\mathtt{a}_{k+1}\ldots\mathtt{a}_n\sqcup\ ] \tag{1}$$

両端の記号 ␣ は，テープが両方向に伸びており，何も書かれていない空白セルがずっと続いていることを暗示している．つまり，機械の動作としては，(1) の両端には，実際には空白 ␣ が延々と伸びているものと思って取り扱う．

$$\cdots\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\mathtt{a}_0\mathtt{a}_1\mathtt{a}_2\ldots\mathtt{a}_{k-1}\ \boxed{q}\Rrightarrow\mathtt{a}_k\mathtt{a}_{k+1}\ldots\mathtt{a}_n\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\cdots$$

■初期状態: まず，我々は TM の入力テープに文字列 $\mathtt{a}_0\mathtt{a}_1\mathtt{a}_2\ldots\mathtt{a}_n$ を書き，機械 TM を動作させる．機械 TM が最初に動作するときは，初期状態 q_{init} であり，ヘッドは文字列の一番左の位置，つまり $\mathtt{a}_0$ の位置に置かれている．

$$[\ \sqcup\ \boxed{q_{\mathrm{init}}}\Rrightarrow\mathtt{a}_0\mathtt{a}_1\mathtt{a}_2\ldots\mathtt{a}_n\sqcup\ ] \tag{2}$$

■状況の遷移: 機械の動作直後は，初期状態 q_{init} にありヘッドは $\mathtt{a}_0$ を読み込んでいる．1 ステップ後には，別の状態 q_0 に変化し，ヘッドは $\mathtt{a}_0$ を別の文字 $\mathtt{b}_0$ に書き換え，左右のどちらかに 1 セルだけ動くことができる．たとえば，今回はヘッドは右に 1 セル動いたとしよう．そうすると，現在の状況は以下によって表される．

$$[\ \sqcup \mathtt{b}_0\ \boxed{q_0} \Rrightarrow \mathtt{a}_1\mathtt{a}_2\dots\mathtt{a}_n\sqcup\ ]$$

この状況の遷移を $\delta(q_{\mathrm{init}}, \mathtt{a}_0) = (q_1, \mathtt{b}_0, \mathtt{right})$ と書く．一般に，機械 TM は，現在の状態 q とヘッドが現在注目している文字 $\mathtt{a}$ のみに依存して，$\delta(q, \mathtt{a})$ によって決定される新たな状況に遷移する．たとえば，現在の状況が (1) で表されており，$\delta(q, \mathtt{a}_k) = (q', \mathtt{b}_k, \mathtt{left})$ であれば，次の時刻で TM の状態は q から q' に変化し，ヘッドはテープ上の文字 $\mathtt{a}_k$ を文字 $\mathtt{b}_k$ に書き換えたあと，1 セル左に動く．つまり，次の時刻での機械 TM の状況は以下のようになる．

$$[\ \sqcup\mathtt{a}_0\mathtt{a}_1\mathtt{a}_2\dots\mathtt{a}_{k-2}\ \boxed{q'} \Rrightarrow \mathtt{a}_{k-1}\mathtt{b}_k\mathtt{a}_{k+1}\dots\mathtt{a}_n\sqcup\ ]$$

また，ヘッドは両端の空白記号 $\sqcup$ のセルまで行き，そこを別の文字に書き換えてもよい．たとえば，現在の状況として，ヘッドが空白記号 $\sqcup$ を読み込んでいるとしよう．

$$[\ \sqcup\mathtt{b}_0\mathtt{b}_1\mathtt{b}_2\dots\mathtt{b}_m\ \boxed{q} \Rrightarrow \sqcup\ ] \tag{3}$$

最初に注意したように，右端の空白記号 $\sqcup$ の更に右にも本来は空白記号が $\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup$ と連なっている．したがって，一つ空白記号を別の文字に置き換えても，次の空白記号が新たに記述される．たとえば $\delta(q, \sqcup) = (q', \mathtt{c}, \mathtt{right})$ のような遷移を考えると，次の時刻での状況は以下のようになる．

$$[\ \sqcup\mathtt{b}_0\mathtt{b}_1\mathtt{b}_2\dots\mathtt{b}_m\mathtt{c}\ \boxed{q'} \Rrightarrow \sqcup\ ] \tag{4}$$

■受理状態: 機械 TM は，受理状態 q_{end} と呼ばれる状態を持つ．受理状態に辿り着いた段階で，機械は計算を停止し，現在のテープに書かれた文字列を出力とする．たとえば，

$$[\ \sqcup\mathtt{c}_0\mathtt{c}_1\mathtt{c}_2\dots\mathtt{c}_{j-1}\ \boxed{q_{\mathrm{end}}} \Rrightarrow \mathtt{c}_j\mathtt{c}_{j+1}\dots\mathtt{c}_s\sqcup\ ] \tag{5}$$

という状況になったとしたら，以後は遷移を行わず，テープ上の文字列 $\mathtt{c}_0\mathtt{c}_1\mathtt{c}_2\dots\mathtt{c}_s$ が出力される．つまり，動作開始時の状況 (2) から遷移を繰り返し，状況 (5) に辿り着いた場合，機械 TM に文字列 $\mathtt{a}_0\mathtt{a}_1\mathtt{a}_2\dots\mathtt{a}_n$ を入力したら文字列 $\mathtt{c}_0\mathtt{c}_1\mathtt{c}_2\dots\mathtt{c}_s$ が出力された，ということである．

受理状態に辿り着く場合，機械 TM は計算を**停止** (*halt*) する，と言う．

定義 1.1. チューリング機械 (*Turing machine*) とは，組 $M = (\Sigma, \sqcup; Q, q_{\mathrm{init}}, q_{\mathrm{end}}; \delta, \mathsf{left}, \mathsf{right})$ で以下の条件を満たすものである．

- Σ は有限集合であり，**アルファベット** (*alphabet*) と呼ばれる．また，$\sqcup \notin \Sigma$ であり，$\sqcup$ は空白記号と呼ばれる．

- Q は有限集合であり，q_{init}, q_{end} は Q の要素である．各 $q \in Q$ は**状態** ($state$) と呼ばれ，特に q_{init} は初期状態，q_{end} は受理状態と呼ばれる．
- $\delta : Q \times (\Sigma \cup \{\sqcup\}) \to Q \times \Sigma \times \{\mathsf{left}, \mathsf{right}\}$ は遷移関数と呼ばれる．

チューリング機械による計算の厳密な定義を与える必要がある．上に記述した機械 TM による計算の遷移を厳密な形で書き下せばよい．

このために，ふたたび現在の状況が (1) である場合を考えよう．遷移が $\delta(q, \mathtt{a_k}) = (r, \mathtt{b_k}, \mathtt{right})$ によって与えられるとする．このとき，チューリング機械の状態は r に変化し，記号 $\mathtt{a_k}$ は $\mathtt{b_k}$ に書き換わり，ヘッドは 1 つ右，つまり $\mathtt{a_{k+1}}$ の位置に移動する．

$$[\ \sqcup \mathtt{a_0 a_1 a_2} \ldots \mathtt{a_{k-1} b_k}\ \boxed{r} \Rrightarrow \mathtt{a_{k+1}} \ldots \mathtt{a_n} \sqcup\ ]$$

もし $\delta(r, \mathtt{b_k}) = (s, \mathtt{c_k}, \mathtt{left})$ であれば，次にチューリング機械の状態は s に変化し，記号 $\mathtt{b_k}$ は $\mathtt{c_k}$ に書き換わり，ヘッドは 1 つ左，つまり $\mathtt{c_k}$ の位置に移動する．

$$[\ \sqcup \mathtt{a_0 a_1 a_2} \ldots \mathtt{a_{k-1}}\ \boxed{s} \Rrightarrow \mathtt{c_k a_{k+1}} \ldots \mathtt{a_n} \sqcup\ ]$$

状況を表す図式をただの文字列とみなすと，一手先の状況は，状態が書かれている箇所とその両隣だけが変化し得る．たとえば，上の 2 回の遷移において，チューリング機械の状態が書かれている箇所の両隣に注目すると，以下のような文字列書換が行われていることが分かる．

$$\begin{aligned} \mathtt{a_{k-1}}\ \boxed{q} \Rrightarrow \mathtt{a_k} \ &\longrightarrow\ \mathtt{a_{k-1} b_k}\ \boxed{r} \Rrightarrow \\ \mathtt{b_k}\ \boxed{r} \Rrightarrow \mathtt{a_{k+1}} \ &\longrightarrow\ \boxed{s} \Rrightarrow \mathtt{c_k a_{k+1}} \end{aligned}$$

そして，チューリング機械の状態が書かれている箇所とその両隣以外の部分の文字列には変化は一切ない．ただし，ヘッドが空白を読み込んでいる場合については注意が必要である．たとえば，状況 (3) から状況 (4) への遷移では，以下のような文字列書換が行われている．

$$\mathtt{b_m}\boxed{q} \Rrightarrow \sqcup\ ] \ \longrightarrow \mathtt{b_m c}\ \boxed{q} \Rrightarrow \sqcup\ ]$$

定義 1.2. チューリング機械 M の書換規則 $\longrightarrow_M$ は以下によって定義される．状態 $q \in Q$ と文字 $\mathtt{a}, \mathtt{b} \in \Sigma$ が与えられたとき，

$$\mathtt{a}\ \boxed{q} \Rrightarrow \mathtt{b} \longrightarrow_M \begin{cases} \boxed{r} \Rrightarrow \mathtt{ab'} & \text{if } \delta(q, \mathtt{b}) = (r, \mathtt{b'}, \mathtt{left}) \\ \mathtt{ab'}\ \boxed{r} \Rrightarrow & \text{if } \delta(q, \mathtt{b}) = (r, \mathtt{b'}, \mathtt{right}) \end{cases}$$

$$\mathtt{a}\ \boxed{q} \Rrightarrow \sqcup\ ] \longrightarrow_M \begin{cases} \boxed{r} \Rrightarrow \mathtt{a c} \sqcup\ ] & \text{if } \delta(q, \sqcup) = (r, \mathtt{c}, \mathtt{left}) \\ \mathtt{ac}\ \boxed{r} \Rrightarrow \sqcup\ ] & \text{if } \delta(q, \sqcup) = (r, \mathtt{c}, \mathtt{right}) \end{cases}$$

$$[\ \boxed{q} \Rrightarrow \sqcup \mathtt{b} \longrightarrow_M \begin{cases} [\ \boxed{r} \Rrightarrow \sqcup \mathtt{cb} & \text{if } \delta(q, \sqcup) = (r, \mathtt{c}, \mathtt{left}) \\ [\ \sqcup \mathtt{c}\ \boxed{r} \Rrightarrow \mathtt{b} & \text{if } \delta(q, \sqcup) = (r, \mathtt{c}, \mathtt{right}) \end{cases}$$

集合 A が与えられたとき，A の要素の有限列を A 上の**語** (*word*) と呼ぶ．A 上の語全体の集合を A^* と書く ($A^{<\omega}$ と書く流儀もある)．ところで，

$$A_M = \Sigma \cup \{\sqcup, [,]\} \cup \{\boxed{q}\Rrightarrow : q \in Q\}$$

と定義すると，チューリング機械 M の各時刻での状況は，A_M 上の語として書かれていることが分かる．したがって，書換規則 $\longrightarrow_M$ は A_M^* 上の 2 項関係，つまり $\longrightarrow_M \subseteq A_M^* \times A_M^*$ である．

- $u \longrightarrow_M v$ であるとき，任意の語 $s, t \in A_M^*$ について，$sut \longrightarrow_M^1 svt$ と書く．
- 各 $s \in \mathbb{N}$ について，語 σ を s 回の書換規則 $\longrightarrow_M$ の適用によって語 τ に書き換えられるとき，$\sigma \longrightarrow_M^s \tau$ と書く．正確には，$\sigma \longrightarrow_M^s \tau$ とは，

  $$\sigma = \sigma_0 \longrightarrow_M^1 \sigma_1 \longrightarrow_M^1 \dots \longrightarrow_M^1 \sigma_{s-1} = \tau$$

  となること $(\sigma_i)_{i<s}$ が存在することとして定義する．
- 語 σ を高々有限回の書換規則 $\longrightarrow_M$ の適用によって語 τ に書き換えられるとき，$\sigma \longrightarrow_M^* \tau$ と書く．つまり，$\sigma \longrightarrow_M^* \tau$ とは，ある $s \in \mathbb{N}$ が存在して，$\sigma \longrightarrow_M^s \tau$ となることである．

つまり，σ がある時刻での M の計算の状況であり，$\sigma \longrightarrow_M^s \tau$ であれば，s ステップ後の M の計算の状況が τ であることを意味する．そして，$\sigma \longrightarrow_M^* \tau$ とは，有限時間経過後に M の計算の状況が τ となることを意味する．混乱の恐れが無い場合は，$\longrightarrow_M^1$ のことも単に $\longrightarrow_M$ と書く．

定義 1.3. チューリング機械 M に文字列 $\sigma = \mathtt{a}_0\mathtt{a}_1 \dots \mathtt{a}_n$ を入力したときの**時刻 s での計算状況** (*configuration at stage s*) とは，次を満たす語 $\tau \in A_M^*$ である．ある $t \leqslant s$ について，

$$[\ \sqcup \boxed{q_{\mathrm{init}}}\Rrightarrow \mathtt{a}_0\mathtt{a}_1 \dots \mathtt{a}_n \sqcup\ ] \longrightarrow_M^t \tau$$

であり，もし $t < s$ ならば，τ は $\boxed{q_{\mathrm{end}}}\Rrightarrow$ を含んでいる．この場合，$M(\mathtt{a}_0\mathtt{a}_1 \dots \mathtt{a}_n)[s] = \tau$ と書く．

つまり，$M(\mathtt{a}_0\mathtt{a}_1 \dots \mathtt{a}_n)[s]$ とは，計算開始から s ステップ後の計算状況を表す．ただし，ある $t < s$ ステップで計算が終了している場合，時刻 t 時点での計算状況を表す，というのが後者の条件が意味するものである．ここで，$M(\mathtt{a}_0\mathtt{a}_1 \dots \mathtt{a}_n)[s] = \tau$ という記法は，各時刻での計算状況が一意に定まることから正当化される．つまり，定義 1.2 を眺めると容易に分かると思うが，$\sigma \in A_M^*$ がチューリング機械 M のある計算状況を表す文字列であれば，$\sigma \longrightarrow_M^1 \tau$ なる $\tau \in A_M^*$ は高々 1 つしか存在しない．よって，M は一価関数として定義されている：

$$(\forall \mathtt{a}_0\mathtt{a}_1 \dots \mathtt{a}_n \in \Sigma^*)(\forall s \in \mathbb{N})(\exists! \tau \in A_M^*)\ M(\mathtt{a}_0\mathtt{a}_1 \dots \mathtt{a}_n)[s] = \tau.$$

定義 1.4. チューリング機械 M が入力 $\sigma = \mathtt{a_0a_1 \ldots a_n} \in \Sigma^*$ に対して計算を**停止** (*halt*) するとは，ある $s \in \mathbb{N}$ について，$M(\sigma)[s]$ が $\boxed{q_{\mathrm{end}}}\Rrightarrow$ を含むことを意味する．言い換えれば，$\boxed{q_{\mathrm{end}}}\Rrightarrow$ を含む語 $\tau \in A_M^*$ で，

$$\mathtt{[\ \sqcup}\boxed{q_{\mathrm{init}}}\Rrightarrow \mathtt{a_0a_1 \ldots a_n \sqcup\ ]} \longrightarrow_M^* \tau$$

となるものが存在することである．この場合，$M(\sigma)\downarrow = \tau$ と書く．そのような τ が存在しない場合，$M(\sigma)\uparrow$ と書く．

さて，チューリング機械の計算状況の書換手続の結果，$M(\sigma)\downarrow = \tau$ となったとしよう．しかし，τ には状態記号 $\boxed{q}\Rrightarrow$, 左端の記号 `[`, 右端の記号 `]`, 空白記号 ␣ などの不要な情報が含まれている．これらの記号を**制御記号**と呼ぶこととしよう．また，我々は Σ の要素についても，どの記号が制御記号かを自由に指定できるとする．文字列 τ から，制御記号を全て取り除いた結果を，τ の**解釈**と呼び，$[\![\tau]\!]$ と書く．

例 1.5. 我々の定義では，チューリング機械はテープに空白記号 ␣ を書くことを許していないため，文字を削除することができない．これは，チューリング機械を定義 1.2 のような有限文字列の書換操作として表現するための都合上のものである．実際には，チューリング機械を文字列書換操作として表現する数学的理由はあまりないので，空白記号の書込みを許すように定義を訂正してもよい．しかし，チューリング機械を文字列書換操作として表現した場合の恩恵もある．このため，たとえば，アルファベット Σ の中に新しい文字 _ を加えて，これを制御記号として指定しよう．こうすれば，出力文字列 τ の解釈 $[\![\tau]\!]$ は，制御記号 _ が全て取り除かれたものとなる．つまり，記号 _ は空白を意味するものとして解釈される．

定義 1.6. チューリング機械 $M = (\Sigma'; Q; \delta)$ が与えられているとする．任意の $\sigma \in \Sigma^*$ について，以下によって $[\![M]\!](\sigma)$ を定義する．

$$[\![M]\!](\sigma) = \begin{cases} [\![M(\sigma)]\!] & \text{if } M(\sigma)\downarrow \\ \text{未定義} & \text{if } M(\sigma)\uparrow \end{cases}$$

この $[\![M]\!]$ をチューリング機械 M の**解釈**と呼ぶ．

関数 $[\![M]\!]$ は Σ^* 全域で定義されるとは限らない．このような関数を Σ^* 上の部分関数と呼ぶ．より一般に，集合 X が与えられたとき，$D \subseteq X^k$ を定義域とする関数 $f : D \to Y$ を X から Y への**部分関数** (*partial function*) と呼ぶ．ただし，計算理論において，定義域 D を知ることは一般的には困難であり，アプリオリに D が与えられるものではない．特に，入力値の領域 X は知っているが，関数の本当の定義域 D（入力領域のうち出力が返ってくる部分）を知る術はない，ということが多々ある．このため，定義域 D に陽に言及することはせず，f が X から Y への部分関

数であることを $f :\subseteq X \to Y$ と表す．たとえば，チューリング機械 $M = (\Sigma'; Q; \delta)$ は部分関数 $[\![M]\!] :\subseteq \Sigma^* \to \Sigma^*$ として解釈される．

定義 1.7. 部分関数 $f :\subseteq \Sigma^* \to \Sigma^*$ が**計算可能** (*computable*) であるとは，あるチューリング機械 M が存在して，任意の語 $\sigma \in \mathrm{dom}(f)$ に対して，$[\![M]\!](\sigma) = f(\sigma)$ となることである．

■自然数上の計算理論: ここまで文字列上の関数について議論してきたが，計算理論においては，自然数上の（多変数）部分関数も重要な考察対象である．自然数上の計算理論は，自然数を文字列でコードすることによって容易に取り扱うことができる．

自然数 $x \in \mathbb{N}$ を 2 進表記したものを $\mathtt{bin}(x)$ と表す．たとえば，$\mathtt{bin}(314) = \mathtt{100111010}$ である．自然数の有限列 $(x_1, x_2, x_3, \ldots, x_k) \in \mathbb{N}^k$ は以下のようにコードしよう．

$$\mathtt{bin}(x_1)\mathtt{,}\mathtt{bin}(x_2)\mathtt{,}\mathtt{bin}(x_3)\mathtt{,}\ldots\mathtt{,}\mathtt{bin}(x_k)$$

したがって，用いるアルファベットは $\Sigma = \{\mathtt{0}, \mathtt{1}, \mathtt{,}, \mathtt{>}, \mathtt{<}\}$ である．

定義 1.8. 部分関数 $f :\subseteq \mathbb{N}^k \to \mathbb{N}$ が**計算可能** (*computable*) であるとは，あるチューリング機械 M が存在して，任意の $(x_1, x_2, \ldots, x_k) \in \mathbb{N}^k$ に対して，$x \in \mathrm{dom}(f)$ かつ $f(x) = y$ ならば，ある $\sigma \in \Sigma^*$ について，

$$[\![M]\!](\mathtt{bin}(x_1)\mathtt{,}\mathtt{bin}(x_2)\mathtt{,}\ldots\mathtt{,}\mathtt{bin}(x_k)) \downarrow = \mathtt{bin}(y)$$

となることである．

混乱の恐れがない場合は，上のような状況のときも $[\![M]\!](x_1, x_2, \ldots, x_k) = y$ と書く．

例 1.9. $f(x) = x + 1$ は計算可能である．チューリング機械 M の状態は $Q = \{q, q', q_{\mathrm{init}}, q_{\mathrm{end}}\}$ からなる．遷移関数は以下によって与えられる．

$$\begin{aligned}
\delta(q_{\mathrm{init}}, \mathtt{a}) &= (q_0, \mathtt{a}, \mathtt{left}) \\
\delta(q_r, \mathtt{a}) &= \begin{cases} (q_r, \mathtt{a}, \mathtt{right}) & \text{if } \mathtt{a} = \mathtt{0} \text{ or } \mathtt{a} = \mathtt{1} \\ (q_\ell, \mathtt{_}, \mathtt{left}) & \text{if } \mathtt{a} = \sqcup \end{cases} \\
\delta(q_\ell, \mathtt{a}) &= \begin{cases} (q_{\mathrm{end}}, \mathtt{1}, \mathtt{left}) & \text{if } \mathtt{a} = \mathtt{0} \\ (q_\ell, \mathtt{0}, \mathtt{left}) & \text{if } \mathtt{a} = \mathtt{1} \\ (q_0, \mathtt{1}, \mathtt{left}) & \text{if } \mathtt{a} = \sqcup \end{cases} \\
\delta(q_0, \mathtt{a}) &= (q_{\mathrm{end}}, \mathtt{_}, \mathtt{left}).
\end{aligned}$$

これ以外の遷移は起きないので任意でよい．実際に，計算の遷移がどのように起きるか見てみよう．たとえば $x = 19$ とすると，$\mathtt{bin}(x) = \mathtt{10011}$ である．このとき，計算は以下のように遷移

する．

$$
\begin{aligned}
&\texttt{[}\sqcup\boxed{q_{\text{init}}}\Rrightarrow\texttt{10011}\sqcup\texttt{]} \longrightarrow_M \texttt{[}\sqcup\texttt{1}\boxed{q_r}\Rrightarrow\texttt{0011}\sqcup\texttt{]} \longrightarrow_M \texttt{[}\sqcup\texttt{11}\boxed{q_r}\Rrightarrow\texttt{011}\sqcup\texttt{]}\\
\longrightarrow_M^*&\texttt{[}\sqcup\texttt{10011}\boxed{q_r}\Rrightarrow\sqcup\texttt{]} \longrightarrow_M \texttt{[}\sqcup\texttt{1001}\boxed{q_\ell}\Rrightarrow\texttt{1\_}\sqcup\texttt{]} \longrightarrow_M \texttt{[}\sqcup\texttt{100}\boxed{q_\ell}\Rrightarrow\texttt{10\_}\sqcup\texttt{]}\\
\longrightarrow_M&\texttt{[}\sqcup\texttt{10}\boxed{q_\ell}\Rrightarrow\texttt{000\_}\sqcup\texttt{]} \longrightarrow_M \texttt{[}\sqcup\texttt{1}\boxed{q_{\text{end}}}\Rrightarrow\texttt{0100\_}\sqcup\texttt{]}
\end{aligned}
$$

$\texttt{bin}(20) = \texttt{10100}$ であるから，$[\![M]\!](19)\downarrow = 20$ である．$x = 2^n - 1$ の形の場合だけ，少し計算の遷移が異なるので，たとえば $x = 15$ としよう．このとき，$\texttt{bin}(x) = \texttt{1111}$ であり，計算は以下のように遷移する．

$$
\begin{aligned}
&\texttt{[}\sqcup\boxed{q_{\text{init}}}\Rrightarrow\texttt{1111}\sqcup\texttt{]} \longrightarrow_M \texttt{[}\sqcup\texttt{1}\boxed{q_r}\Rrightarrow\texttt{111}\sqcup\texttt{]} \longrightarrow_M \texttt{[}\sqcup\texttt{11}\boxed{q_r}\Rrightarrow\texttt{11}\sqcup\texttt{]}\\
\longrightarrow_M^*&\texttt{[}\sqcup\texttt{1111}\boxed{q_r}\Rrightarrow\sqcup\texttt{]} \longrightarrow_M \texttt{[}\sqcup\texttt{111}\boxed{q_\ell}\Rrightarrow\texttt{1\_}\sqcup\texttt{]} \longrightarrow_M \texttt{[}\sqcup\texttt{11}\boxed{q_\ell}\Rrightarrow\texttt{10\_}\sqcup\texttt{]}\\
\longrightarrow_M&\texttt{[}\sqcup\texttt{1}\boxed{q_\ell}\Rrightarrow\texttt{100\_}\sqcup\texttt{]} \longrightarrow_M \texttt{[}\sqcup\boxed{q_\ell}\Rrightarrow\texttt{1000<}\sqcup\texttt{]} \longrightarrow_M \texttt{[}\boxed{q_\ell}\Rrightarrow\sqcup\texttt{0000\_}\sqcup\texttt{]}\\
\longrightarrow_M&\texttt{[}\boxed{q_0}\Rrightarrow\sqcup\texttt{10000\_}\sqcup\texttt{]} \longrightarrow_M \texttt{[}\boxed{q_{\text{end}}}\Rrightarrow\sqcup\texttt{\_10000\_}\sqcup\texttt{]}
\end{aligned}
$$

$\texttt{bin}(16) = \texttt{10000}$ であるから，$[\![M]\!](15)\downarrow = 16$ である．一般に $[\![M]\!](x) = x + 1$ となっていることを確認することは難しくない．

演習問題 1.10. 自然数上の部分的減法 $\dot{-}$ を $x \dot{-} y = \max\{0, x - y\}$ によって定義する．つまり，

$$
x \dot{-} y = \begin{cases} x - y & \textit{if } x \geqslant y \\ 0 & \textit{otherwise.} \end{cases}
$$

このとき，$f(x, y) = x \dot{-} y$ が計算可能であることを示せ．

1.2 文字列書換系とモノイドの表示

定義 1.11. 文字列書換系 (*string rewriting system*) とは，集合 A と語上の関係 $\to \subseteq A^* \times A^*$ の対 $(A, \to)$ である．もし A と $\to$ が共に有限であるならば，$(A, \to)$ を有限文字列書換系と呼ぶ．

文字列書換系における $\to$ は，**書換規則** (*rewrite rules*) をリストアップしたものと考える．各規則 $u \to v$ は，与えられた語が u を部分語として含んでいるならば，u の部分を v に書き換えることができることを意味する．

例 1.12. チューリング機械 M について，定義 1.2 の $(A_M, \longrightarrow_M)$ は有限文字列書換系である．

文字列書換系は，**半群の表示** (*presentation of semigroup*)，より正確にはモノイドの表示と密接に関係している．

定義 1.13. 半群 (*semigroup*) とは，結合的な 2 変数関数 $* : S^2 \to S$ の定義された集合 S である．

つまり，$*$ は以下の結合法則を満たす．

$$\forall a, b, c \in S\ ((a * b) * c = a * (b * c)).$$

例 1.14. 集合 A 上の語全体 A^* について，積 $*$ を語の結合，つまり $u * v = uv$ によって定義すると，これは半群をなす．

定義 1.15. 半群 S 上の 2 項関係 $R \subseteq S \times S$ に関する以下の性質を考える．

$$\begin{array}{rl} \text{反射律} & (\forall s \in S)\ sRs \\ \text{推移律} & (\forall s, t, u \in S)\ sRt \wedge tRu \to sRu \\ \text{積の保存} & (\forall s, t, u \in S)\ sRt \to (s * u)R(t * u) \\ & (\forall s, t, u \in S)\ sRt \to (u * s)R(u * t) \end{array}$$

半群 S 上の与えられた 2 項関係 R について，R を含み，反射律，推移律，積の保存を満たす最小の 2 項関係を R^* と書く．

例 1.16. チューリング機械 M について，$(A_M, \longrightarrow_M)$ を定義 1.2 の文字列書換系とすると，例 1.14 のように，A_M^* は半群をなし，$\longrightarrow_M$ は半群 A_M^* 上の 2 項関係となる．このとき，第 1.1 節で定義した $\longrightarrow_M^*$ と定義 1.15 の $\longrightarrow_M^*$ は一致する．

例 1.17. より一般に，$(A, \to)$ が文字列書換系であれば，以下によって定義される $\to^*$ は，定義 1.15 の $\to^*$ と一致する．

- $u \to v$ であるとき，任意の語 $s, t \in A^*$ について，$sut \to^1 svt$ と書く．この遷移 $\to^1$ は**一手書換** (*one-step rewriting*) と呼ばれる．
- 語 s を高々有限回の書換規則の適用によって語 t に書き換えられるとき，$s \to^* t$ と書く．正確には，$s \to^* t$ とは，$s = t$ または $s = s_0 \to^1 s_1 \to^1 \cdots \to^1 s_n = t$ となることとして定義する．つまり，$\to^*$ は一手書換 $\to^1$ の推移閉包である．

定義 1.18. 半群 S 上の 2 項関係 $R \subseteq S \times S$ が**合同関係** (*congruence relation*) であるとは，R が反射律，推移律，積の保存，および以下の対称律を満たすことである．

$$(\forall s, t \in S)\ sRt\ \to\ tRs.$$

例 1.19. 半群 S 上の与えられた 2 項関係 R について，sRt または tRs であるとき $sR_{\mathrm{sym}}t$ と定義する．また，例 1.17 のように R_{sym}^* を R_{sym} の推移閉包として定義する．このとき，$x \equiv_R y$ を xR_{sym}^*y によって定義すれば，$\equiv_R$ は R を含む最小の合同関係である．

定義 1.20. モノイド (*monoid*) とは，単位元 $\varepsilon \in S$ を持つ半群 S である．つまり，ε は以下の性質を満たす．

$$\forall a \in S\ (a * \varepsilon = \varepsilon * a = a)$$

例 1.21. 自然数上の加法 $+$ と零 0 を考えると，$(\mathbb{N}, +, 0)$ はモノイドをなす．$n \times n$ 整数行列全体の集合を $M_n(\mathbb{Z})$ と書き，$M_n(\mathbb{Z})$ 上の積 $\cdot$ と単位行列 E を考えると，$(M_n(\mathbb{Z}), \cdot, E)$ はモノイドを

なす．集合 X 上の関数 $f : X \to X$ の合成 $\circ$ と恒等写像 id を考えると，$(X^X, \circ, \mathrm{id})$ はモノイドをなす．

例 1.22. 集合 A 上の語全体 A^* について，積 $*$ を語の結合，ε を空語とすれば，これはモノイドをなす．これを A 上の**自由モノイド** (*free monoid*) と呼ぶ．

定義 1.23. A^* を A 上の自由モノイドとする．関係 $R \subseteq A^* \times A^*$ が与えられたとき，$\equiv_R$ を例 1.19 の合同関係とする．このとき，$\langle A \mid R \rangle$ を商モノイド $M = A^*/\equiv_R$ の**表示** (*presentation*) と呼ぶ．もし A と R が共に有限であれば，$\langle A \mid R \rangle$ を M の**有限表示** (*finite presentation*) と呼ぶ．

文字列書換系とモノイドの表示は数学的に同一であり，同様に，有限文字列書換系はモノイドの有限表示と同一である．$(A, \to)$ を文字列書換系とすると，$\to$ は自由モノイド A^* 上の 2 項関係と思える．このとき，$\equiv$ を例 1.19 のような $\to$ を含む最小の合同関係とする．商モノイド $M = A^*/\equiv$ を考えると，各 $[s], [t] \in M$ について次が成立する．

$$[s] = [t] \iff s \equiv t \iff (\exists (s_i)_{i \leqslant k})(\exists (\sim_i)_{i \leqslant k})\ s = s_0 \sim_1 s_1 \sim_2 \cdots \sim_k s_k = t.$$

ここで $\sim_i \in \{\leftarrow, \to\}$ である．つまり，$[s] = [t]$ とは，s と t が $\leftarrow$ または $\to$ の有限ステップで繋がっているということである．同様に，群の表示もある種の文字列書換系と思うことができる．

定義 1.24. 群 (*group*) とは，任意の $x \in G$ が逆元 $x^{-1} \in G$ を持つようなモノイド G である．集合 S が生成する**自由群** (*free group*) F_S は以下によって定義される．まず，各 $s \in S$ 毎に新たな記号 s^{-1} を用意し，$S^{-1} = \{s^{-1} : s \in S\}$ と定義する．このとき，各 $s \in S$ 毎に以下の書換規則を定義する．

$$ss^{-1} \to_S \varepsilon \qquad s^{-1}s \to_S \varepsilon$$

ここで ε は空語を表す．このとき，文字列書換系 $\langle S \cup S^{-1} \mid \to_S \rangle$ から定義される商モノイドのことを S が生成する自由群と呼び，F_S と書く．

定義 1.25. 集合 S と語上の関係 $R \subseteq (S \cup S^{-1})^*$ が与えられたとき，N_R を R が生成する F_S の正規部分群とする．このとき，$\langle S \mid R \rangle$ を商群 $G = F_S/N_R$ の**表示** (*presentation*) と呼ぶ．もし S と R が共に有限であれば，$\langle S \mid R \rangle$ を G の有限表示 (*finite presentation*) と呼ぶ．

上の定義が述べていることは，各 $s \in R$ 毎に以下の書換規則を考えることに相当する．

$$s \to_R \varepsilon$$

このとき，文字列書換系 $\langle S \cup S^{-1} \mid \to_S \cup \to_R \rangle$ が定義する商モノイドが，$\langle S \mid R \rangle$ によって表示される群 $G = F_S/N_R$ と考えてよい．

1.3 チューリング機械を用いた関数の実装

チューリング機械を使ったプログラムを書くのは極めて面倒であるし，どのような関数がチューリング機械によって計算できるか一見では明らかではない．まず，チューリング機械でのプログラミングの難しさは逐次アクセスメモリ（つまり，端から順にしかデータにアクセスできないメモリ）を取り扱っている部分による所が大きい．多くの現代人はランダムアクセスメモリ（任意の場所のデータにアクセスできる）の方に慣れ親しんでいることが，その難しさに拍車を掛ける．

ここでは，チューリング機械による計算に近く，また，ランダムアクセス機械でのプログラミングにも比較的近い，多テープ・チューリング機械を用いた疑似プログラミング言語を導入する．

■多テープ・チューリング機械: **k-テープ・チューリング機械**とは，k 個のテープを持つチューリング機械である．遷移関数が $\delta : Q \times (\Sigma \cup \{\sqcup\})^k \to Q \times (\Sigma \times \{\mathsf{left}, \mathsf{stay}, \mathsf{right}\})^k$ によって与えられるという点以外は，チューリング機械と変わらない．つまり，各テープ毎にヘッドがあり，現在の状態と各ヘッドが読み込んでいる文字たちの組み合わせによって，次の時点での状況が決定される．ただし，入力は第 0 テープに記述され，第 0 テープに記述したものが出力されるものとする．他のテープは計算過程で用いるのみである．また，簡便のためにヘッドを動かさない処理である stay も加えておくこととする．

定理 1.26. 任意の自然数 $k \geqslant 1$ について，1 テープ・チューリング機械と k テープ・チューリング機械の計算能力は等しい．

演習問題 1.27. 定理 *1.26* を証明せよ．

■疑似プログラミング言語 TURING　これから，多テープ・チューリング機械を疑似プログラミング言語として取り扱う．まず，各テープは，文字列を格納する変数，すなわち String 型の変数だと考える．第 k 番目のテープが表す変数を tape[k] と書く．実際の実装には，まずプログラムを与えた後，プログラムに現れる変数の種類の数だけテープを用意してチューリング機械を構築する，という形となる．特に，プログラムを 1 つ書く毎に，チューリング機械が 1 つ必要であることに注意する．

チューリング機械の状態は，プログラムの行番号に対応する．したがって，チューリング機械の状態の集合 Q のサイズとはプログラムの行数である．ただし，我々は今後マクロを定義し，プログラムの記述にマクロを用いるため，表面上のプログラムの行数と，機械の処理上の本来のプログラムの行数にずれが生じる．実際には，チューリング機械の状態は，マクロを全て基本コマンドに置き換えてプログラムを書き直したときの行番号に対応する．

■基本コマンド:　まず，疑似プログラミング言語 TURING における基本コマンドを導入する．以下のコマンドがチューリング機械によって実装できることは明らかであろう．

`move_left[i]; % 第 i ヘッドを 1 セル左に動かす`
`move_right[i]; % 第 i ヘッドを 1 セル右に動かす`
`tape[i](head):=a; % 第 i テープのヘッド位置の記号を a に書き換える．`

このような基本コマンドを用いて，プログラムを記述するが，プログラムには行番号が割り当てられる．特に言及が無い場合，プログラムの上の行から順に実行していく．これは，言及が無ければ，第 n 状態 q_n の直後の時刻では第 $n+1$ 状態 q_{n+1} となっていることを意味する．しかし，チューリング機械の基本動作としては，状態 q_n の直後に q_{n+1} 以外の状態になることも許される．これは，プログラム中に goto 文が使えることを意味する．

`goto k; % プログラムの第 k 行へ移動する．`

また，チューリング機械の基本動作は，ある種の条件分岐文を実行するものである．まず，`tape[i](head)=a` を**原子式**と呼ぶ．これは，第 i テープのヘッド位置に書かれている記号が a であることを意味する．原子式 L0, L1, ..., Li と基本コマンド C と D が与えられたとき，以下のコマンドも用いることができる．

```
if L0 and L1 and ... and Li then C else D;
% もし L0,L1,...,Li が全て真ならば，コマンド C を実行する．
% もし L0,L1,...,Li のどれかが偽ならば，コマンド D を実行する．
```

これは遷移関数が k 個のヘッドが読み込んでいる値の組み合わせに依存して，次の時刻での遷移を決定できることに対応する．記号の種類が有限個であるから，基本条件分岐コマンドを入れ子にして用いることによって，原子式 L の代わりに原子式の有限ブール結合を用いてもよいことも分かる．同様に，C と D の部分には，複数のコマンドを配置してもよい．つまり，C と D の部分は単独のコマンドだけでなく，言語 TURING の任意のプログラムを記述できる．

以上に述べた 4 種類のコマンドを**言語 TURING の基本コマンド**と呼ぶ．言語 TURING における計算とは，これらの 4 種の基本コマンドを組み合わせて書かれたプログラムによる計算である．ここで，プログラムが何も書かれていない行を読み込んだとき，計算は終了する．

■応用コマンド: 十分大きな k について，コマンド goto k は計算を停止する（受理状態に移行する）命令だと思える．また，プログラムの第 k 行に goto k+1 と記述すれば，これは何もしない命令と同様である．これは，以下のコマンドが言語 TURING で実装できることを意味する．

```
end; % 計算を停止する．
nothing; % 何もしない．
```

この言語 TURING の計算能力を理解するために，まず，条件分岐と goto 文を組み合わせれば WHILE ループが作れることを確認しよう．つまり，言語 TURING において，以下のコマンドを利用できる．

```
while tape[i](head)=a { P };
```
% 第 i テープのヘッドの位置の記号が a である間はずっとプログラム P を繰り返す．

このコマンドは以下のプログラムによって実装できる．

```
00  if tape[i](head)=a then P else end;
01  goto 00;
```

同様の方法でループコマンド `while tape[i](head)!=a { P }` も実装できる．これは，第 i テープのヘッドの位置の記号が a になるまで，ずっとプログラム P を繰り返す命令である．

■実装の簡易化のための約束事: これから，様々な関数を言語 TURING の基本コマンドを用いて実装していこう．実装の簡便性のために，各コマンドの実行開始時点では，テープ上には常に開始記号 > と終了記号 < が書かれているとする．つまり，各テープに書かれている文字は以下のようなものである．

$$\ldots ␣␣␣ a_0 a_1 \ldots a_i > b_0 b_1 \ldots b_j < c_0 c_1 \ldots c_k ␣␣␣ \ldots$$

ここで，$a_0, \ldots a_i, b_0, \ldots, b_j$ の中に記号 $>, <, ␣$ は含まれない．このとき，このテープには $b_0 b_1 \ldots b_j$ という文字列が格納されているものと思って取り扱う．

また，各コマンドの実行終了時点でも，テープ上には必ず開始記号 > と終了記号 < が書かれており，ヘッドは開始記号 > と終了記号 < の間のどこかに位置しているような実装を行う．

■ヘッド位置初期化コマンド `init_head[i];` ヘッドを開始記号 > の位置に自動的に戻すコマンドがあれば便利である．これは，開始記号 > を読み込むまでヘッドを左に動かすチューリング機械の遷移によって容易に実現できる．

```
00  while tape[i](head)!=> {
01      move_left[i];
02  }
```

■条件分岐コマンド `if tape[i]=0 then { P } else { Q };` 変数 tape[i] に格納された文字列が >0< であれば，プログラム P を実行し，さもなくばプログラム Q を実行する命令である．第 i ヘッドを開始位置 > に戻し，右 2 つを読み込んで 0< であれば P を実行し，さもなくば Q を実行することによって実装できる．

```
00  init_head[i];
01  move_right[i];
02  if tape[i](head)=0 then move_right[i] else goto 04;
03  if tape[i](head)=< then { P; end; } else goto 04;
04  Q;
```

■2 進部抽出コマンド tape[i]:=binary(tape[j]); 変数 tape[j] に格納された文字列から 2 進部の情報 binary(tape[j]) を読み出し，その値を変数 tape[i] に代入する命令である．たとえば，第 j テープに書かれた文字列が >001101abb< であれば，第 i テープの文字列を >001101< と書き換える．

実装方法としては，第 i と第 j ヘッドを開始位置に戻した後，第 j ヘッドが 0 と 1 以外の記号を読み込むまで両方のヘッドを右に動かす．0 と 1 以外の記号を読み込んだら，第 i ヘッドはその位置に終了記号 < を書き込み，両方のヘッドを 1 つずつ左に動かしながら，第 j テープの内容を第 i テープにコピーしていけばよい．

```
00  init_head[i];  init_head[j];
01  move_right[i];  move_right[j];
02  while tape[j](head)=0 or tape[j](head)=1 {
03      move_right[i];  move_right[j];
04  }
05  tape[i](head):=<;
06  move_left[i];  move_left[j];
07  while tape[j](head)=0 or tape[j](head)=1 {
08      tape[i](head):=tape[j](head);
09      move_left[i];  move_left[j];
10  }
```

■自然数部抽出コマンド tape[i]:=int(tape[j]); 変数 tape[j] に格納された文字列から自然数情報 int(tape[j]) を読み出し，その値を変数 tape[i] に代入する命令である．たとえば，第 j テープに書かれた文字列が >001101abb< であれば，これは 001101 が表す自然数 13 と考え，bin(13) = 1101 なので，第 i テープの文字列を >1101< と書き換える．

■インクリメント tape[i]++; 変数 tape[i] に格納されている自然数値を 1 加算する命令である．

```
00  init_head[i];
01  tape[i]:=int(tape[i]);
02  while tape[i](head)!=< {
03      move_right[i];
04  }
05  move_left[i];
06  while tape[i](head)!=> {
07      if tape[i](head)=1 then {
08          tape[i](head):=0;
```

```
09          move_left[i];
10      }
11      else if tape[i](head)=0 then {
12          tape[i](head):=1; end;
13      }
14      else if tape[i](head)=> then {
15          tape[i](head):=1;
16          move_left[i];
17          tape[i](head):=>; end;
18      }
19  }
```

■デクリメント tape[i]--; 変数 tape[i] に格納されている自然数値を 1 減算する命令である．ヘッドを一旦，終了記号 < の位置まで動かした後，徐々に左に戻していく．もし記号 0 が現れたら 1 に書き換えて左に動き，記号 1 が現れた時点で 0 に書き換えて計算を停止する．ただし，現れた 1 の左が開始記号 > である場合，この >1 を #> に書き換えて計算を停止する．

```
00  init_head[i];
01  tape[i]:=int(head[i]);
02  if tape[i]=0 then end; else nothing;
03  while tape[i](head)!=< {
04      move_right[i];
05  }
06  move_left[i];
07  while tape[i](head)!=> {
08      if tape[i](head)=0 then {
09          tape[i](head):=1;
10          move_left[i];
11      } else nothing;
12      if tape[i](head)=1 then {
13          move_left[i];
14          if tape[i](head)=> then {
15              tape[i](head):=#;
16              move_right[i];
17              tape[i](head):=>; end;
18          } else {
19              move_right[i];
```

```
20              tape[i](head):=0; end;
21          }
22      }
23  }
```

■ループコマンド `for tape[i] { P }`　これは変数 `tape[i]` に格納されている文字列が表す自然数の回数だけ，プログラム `P` を実行する命令である．

```
00  tape[z]:=int(tape[i]);
01  while tape[z]!=0 {
02      P;
03      tape[z]--;
04  }
```

■書換コマンド `tape[i](tape[j]):=a;`　これは変数 `tape[i]` の `int(tape[j])` 文字目の値を `a` に書き換える命令である．これは以下のように実装する．

```
00  init_head[i];
01  for tape[j] {
02      move_right[i];
03  }
04  tape[i](head):=a;
```

同様に `tape[i](tape[j]):=tape[k](tape[l]);` のような書換コマンドも実装できる．これは，これは変数 `tape[i]` の `int(tape[j])` 文字目の値を変数 `tape[k]` の `int(tape[l])` 文字目の値に書き換える命令である．

■文字列検索コマンド `tape[i]:=IsSubstring(tape[j],tape[k]);`　変数 `tape[j]` に格納された文字列が変数 `tape[k]` に格納された文字列の部分列であれば，`tape[j]` が `tape[k]` の何文字目から出現するかを `tape[i]` に書き込み，さもなくば空列を `tape[i]` に書き込む命令である．たとえば，`tape[j]` と `tape[k]` に格納されている文字列がそれぞれ `>ab<` と `>cbabc<` であれば，$\mathtt{bin}(2) = 10$ であるから，`tape[i]` には `>10<` と書き込む．`tape[j]` と `tape[k]` に格納されている文字列がそれぞれ `>abc<` と `>cbab<` であれば，条件を満たさないから，`tape[i]` には `><` と書き込む．

実装のために，補助的なテープを 2 つ用意する．まず，補助変数 `tape[z]` には，まず `tape[k]` に書かれている文字列の長さを書き込む．

```
00  tape[z]:=0;
01  while tape[k](head)!=< {
02      tape[z]++;
```

```
03  }
```

各 $\ell <$ tape[z] について，tape[j](n) = tape[k]$(\ell + n)$ の照合をしていけばよい．現在確認中の ℓ を格納するための変数として，tape[y] を用いる．tape[y] $= \ell$ のとき，まず第 k ヘッドを ℓ セル右に動かす．その後，第 j ヘッドと第 k ヘッドが読み込んでいる値が一致するかどうかを 1 セルずつ右に動かしながら確認していく．途中で間違った値が見つかったら，変数 tape[i] に >0< を書き込み，計算を終了する．さもなくば，第 j ヘッドが終了記号 < を読み込んだ時点で，変数 tape[i] に >bin(ℓ)< を書き込み，計算を終了する．具体的な実装は次によって与えられる．

```
04  tape[y]:=0;
05  for tape[z] {
06      init_head[k];
07      for tape[y] { move_right[k]; }
08      init_head[j];
09      while tape[j](head)!=< {
10          if tape[j](head)=tape[k](head) then {
11              move_right[j]; move_right[k];
12          } else {
13              tape[i]:=0; goto 17;
14          }
15      }
16      tape[i]:=tape[y]; end;
17      tape[y]++;
18  }
```

1.4 万能チューリング機械と停止問題

チューリング機械という計算モデルは，物理的に見れば，計算する関数毎に機械を作っている．たとえば，第 1.3 節の疑似プログラミング言語 TURING について，一つ一つのプログラムがそれぞれ多テープ・チューリング機械なのである．したがって，100 個のプログラムを書くということは，100 個のチューリング機械を用意するということである．

しかし，我々の世代のコンピュータは，「たった 1 つの機械」の中で，プログラムを書くことによって，全ての計算可能関数を実装できる[*1]．このようなコンピュータは，数学的には**万能チューリング機械** (*universal Turing machine*) と呼ばれるものである．1930 年代，そのような機械がまだ存在しなかった時代，チューリングは万能機械の概念を定式化し，理論的にその存在を示した．

[*1] あるいは，「たった 1 つの機械」にソフトウェアをインストールすることによって，無数のアプリケーションを実行できる．

このようにして，我々の世代が用いているようなコンピュータの到来をチューリングは予言したのである．

チューリング機械 U が**万能** (*universal*) とは，任意のチューリング機械 M に対して，

$$(\exists e \in \Sigma^*)(\forall \sigma \in \Sigma^*)\ [\![U]\!](e,\sigma) = [\![M]\!](\sigma).$$

現代的に言えば，U が我々の眼前にあるコンピュータであり，e というものは，機械 M の計算をシミュレートするプログラムである．我々のコンピュータ U の中でプログラム e に文字列 σ を入力すれば，$[\![M]\!](\sigma)$ という計算結果が得られる．現代文明の恩恵を享けている全ての人は，次の定理を体で知っているが，ここでは頭で理解することを目指そう．

定理 1.28. 万能チューリング機械が存在する．

Proof. 有限文字列書換系 $(\Sigma, \to)$ について，$\to$ は有限集合であるから，$\{u_i \to v_i : i \leqslant n\}$ のようにリストアップされていると仮定する．このとき，書換規則 $\to$ を以下の文字列でコードする．

$$u_0\texttt{@}v_0\texttt{|}u_1\texttt{@}v_1\texttt{|}u_2\texttt{@}v_2\texttt{|}\ldots\texttt{|}u_n\texttt{@}v_n$$

この文字列を $\texttt{code(}\to\texttt{)}$ と書く．次のような多テープ・チューリング機械 U を構成する．任意のチューリング機械 M に対して，$\longrightarrow_M$ を定義 1.2 の書換規則とすると，

$$[\![U]\!](\texttt{code(}\longrightarrow_M\texttt{),}\mathtt{a_0a_1\ldots a_n}) = [\![M]\!](\mathtt{a_0a_1\ldots a_n}).$$

チューリング機械 U のおおまかな構成は以下による．まず，変数 tape[0] には，最初は初期状況を表す文字列 $\texttt{[}\sqcup\boxed{q_{\text{init}}}\Rrightarrow \mathtt{a_0a_1\ldots a_n}\sqcup\texttt{]}$ を格納しておく．各 $u \to v$ について，変数 tape[0] に格納された文字列 w について，u が w の部分列になっているか判定する．そうであれば，w の u の部分を v に変えた列が表す数を変数 tape[0] に代入する．この手続を繰り返し，受理状態 $\boxed{q_{\text{end}}}\Rrightarrow$ を含む文字列が得られたら，制御記号の部分を消した残りの文字列を出力して，計算を終了する．

具体的な実装としては，以下の手続きを実行する TURING プログラムのコードを書けばよい．

1. まず，入力文字列の # より左側の文字列を変数 tape[r] に代入し，tape[0] からは消す．
2. tape[0] の文字列を $\texttt{[}\sqcup\boxed{q_{\text{init}}}\Rrightarrow \mathtt{a_0a_1\ldots a_n}\sqcup\texttt{]}$ に書き換える．
3. tape[r] の @ の数をカウントして，変数 tape[z] に代入する．
4. 以下の手続きを tape[z] に格納された自然数の回数だけ繰り返す．
5. tape[y] = 0 からスタートする．
6. 第 r ヘッドを tape[y] の個数分 | を読むまで動かす．
7. 次の @ が現れるまで文字列を tape[x] に複製する．そこから，| が現れるまでの文字列を tape[w] に複製する．
8. IsSubstring(tape[x],tape[0]) を実行し，tape[x] が tape[0] の部分列になっているか調べる．

9. tape[x] が tape[0] の部分列なら，その部分を tape[w] に書き換える．3 に戻る．これを文字列内に $\boxed{q_{\text{end}}}\Rightarrow$ が現れるまで繰り返す．
10. tape[x] が tape[0] の部分列でないなら，tape[y]++ して，5 に戻る．
11. tape[y] が tape[z] を越えたら，計算を終了する．
12. $\boxed{q_{\text{end}}}\Rightarrow$ が現れたら，文字列部分だけ抽出して，計算を終了する．

定理 1.26 より，多テープ・チューリング機械の計算をシミュレートする 1-テープ・チューリング機械が存在するから，定理は示された． □

定理 1.28 の万能チューリング機械 U について，以後，各 $e,n\in\Sigma^*$ に対して，$[\![e]\!](n)=U(e,n)$ と定義する．$e\in\Sigma^*$ があるチューリング機械 M のコードである場合，すなわち $e=\mathsf{code}(\longrightarrow_M)$ である場合，チューリング機械 M が計算する部分関数 $[\![M]\!]:\subseteq\Sigma^*\to\Sigma^*$ (定義 1.6 を参照せよ) と部分関数 $[\![e]\!]:\subseteq\Sigma^*\to\Sigma^*$ は一致する．

万能チューリング機械の存在定理 1.28 の帰結の 1 つとして，以下の**パラメータ定理** (*parameter theorem*) または smn 定理と呼ばれるものを証明できる．

定理 1.29. 次のような計算可能関数 $s:\Sigma^*\to\Sigma^*$ が存在する．任意の $e,u,v\in\Sigma^*$ に対して，

$$[\![s(e,u)]\!](v)=[\![e]\!](u,v).$$

Proof. U を定理 1.28 の万能チューリング機械とすると，$[\![U]\!](e,u,v)=[\![e]\!](u,v)$ である．このとき，$M_{e,u}$ を入力 v に対して，$U(e,u,v)$ の計算をシミュレートするチューリング機械とする．つまり，$[\![M_{e,u}]\!](v)=[\![U]\!](e,u,v)$ である．与えられた e,u から $M_{e,u}$ のコードは容易に計算可能であるから，s をそれを計算する関数とする．このとき，

$$[\![s(e,u)]\!](v)=[\![M_{e,u}]\!](v)=[\![U]\!](e,u,v)=[\![e]\!](u,v)$$

を得る．よって，定理は示された． □

パラメータ定理 1.29 は計算機科学における**カリー化** (*currying*) と呼ばれる操作に対応する．

■決定問題: 計算理論における重要な考察対象の 1 つは，決定問題というタイプの問題が計算可能に解けるかどうか，という点である．たとえば，与えられた 2 進文字列が素数を表しているかどうかを尋ねる問題 Prime を考えよう．もし，$\sigma\in\{0,1\}^*$ が素数を表しているなら 1，さもなくば 0 を返すチューリング機械が作れるならば，素数判定問題 Prime は計算可能に解けると言えるだろう．

このように，**決定問題** (*decision problem*) とは，各文字列 σ に対して真偽が指定された問題である．数学的には，これは集合 $Q\subseteq\Sigma^*$ またはその特性関数 $\chi_Q:\Sigma^*\to\{0,1\}$ と同一視される．た

とえば，上の素数判定問題 Prime は，集合 $P = \{\mathtt{bin}(p) : p \text{ は素数}\}$ または，その特性関数

$$\chi_P(\mathtt{bin}(p)) = \begin{cases} 1 & \text{if } p \text{ は素数} \\ 0 & \text{otherwise} \end{cases}$$

として扱われる．以後は，特性関数の方もわざわざ χ_Q とは書かず，単に Q と表す．つまり，同じ記号 Q が，集合 $Q \subseteq \Sigma^*$ とその特性関数 $Q : \Sigma^* \to \{0,1\}$ の両方を表す．このような同一視をするので，「決定問題 Q が計算可能な方法で解ける」と言う代わりに，単に「決定問題 Q が計算可能である」と言う．

ところで，自然数を扱う際に毎回わざわざ $\mathtt{bin}(n)$ のように書くのは面倒である．自然数と有限アルファベット上の語を同一視してしまっても計算論的には全く影響はないので，以後は特に言及せずに $n \in \mathbb{N}$ と $\mathtt{bin}(n) \in \{\mathtt{0},\mathtt{1}\}^*$ を同一視して取り扱うこととする．

この 2 つの約束事は以後ずっと用いるので，忘れないようにしよう．

- (集合と特性関数の同一視) $n \in Q$ を $Q(n) = 1$ と表し，$n \notin Q$ を $Q(n) = 0$ と表す．
- (自然数と語の同一視) 自然数 $n \in \mathbb{N}$ とその 2 進表記 $\mathtt{bin}(n) \in \{\mathtt{0},\mathtt{1}\}^*$ は同一視する．

演習問題 1.30. 素数判定問題 Prime が計算可能であることを示せ．

定義 1.4 において，計算が停止する（受理状態に到達する）ことを下矢印 $\downarrow$ で表し，そうでない場合，上矢印 $\uparrow$ で表していたことを思い出そう．機械 M の**停止問題** (*halting problem*) とは，入力 n に対して，入力 n に対する M の計算が停止するか否かを判定する問題である．

$$\mathsf{Halt}_M(n) = \begin{cases} 1 & \text{if } [\![M]\!](n) \downarrow \\ 0 & \text{if } [\![M]\!](n) \uparrow . \end{cases}$$

定理 1.28 の万能チューリング機械 U について，$\mathsf{Halt} = \mathsf{Halt}_U$ と定義する．つまり，停止問題 Halt とは，

> 与えられたチューリング機械 M のコード e と入力 n に対して，入力 n に対する M の計算が停止するか否かを判定せよ

という問題である．この停止問題 $\mathsf{Halt} = \mathsf{Halt}_U$ こそが，人類が発見した最初の計算不可能な問題であり，そして計算可能性理論の始まりを告げる定理である．

定理 1.31. 停止問題 Halt は計算不可能である．

Proof. M を任意のチューリング機械とする．このとき，次のようなチューリング機械 N を考える．入力 $\sigma \in \Sigma^*$ が与えられたら，$M(\sigma,\sigma)$ の計算をシミュレートする．もし，$M(\sigma,\sigma)$ の計算が

停止し，その出力が 0 を意味しているときに限り，$N(\sigma)$ は計算を停止し，適当な文字列を出力する．それ以外の場合は，計算を停止しない．つまり，$N :\subseteq \Sigma^* \to \Sigma^*$ は次の部分関数を計算する．

$$[\![N]\!](\sigma) = \begin{cases} \downarrow & \text{if } [\![M]\!](\sigma,\sigma)\downarrow = 0 \\ \uparrow & \text{otherwise.} \end{cases}$$

いま，$e = \mathsf{code}(\longrightarrow_N)$ とする．このとき，

$$\begin{aligned} [\![M]\!](e,e) = 0 &\iff [\![N]\!](e)\downarrow \iff [\![e]\!](e)\downarrow \iff \mathsf{Halt}(e,e) = 1 \\ [\![M]\!](e,e) \neq 0 &\iff [\![N]\!](e)\uparrow \iff [\![e]\!](e)\uparrow \iff \mathsf{Halt}(e,e) = 0. \end{aligned}$$

どちらにせよ，$[\![M]\!](e,e) \neq \mathsf{Halt}(e,e)$ であるから，チューリング機械 M は停止問題を計算しない．いま，M は任意であったから，停止問題が計算不可能であることが導かれる． □

ここで，定理 1.31 は単に計算不可能な問題の存在を示しているという以上の情報を含んでいる．実際，計算可能な関数の存在を示すというだけであれば，万能チューリング機械の議論などは全くもって不必要である．たとえば，高々可算個のチューリング機械しか存在しない一方，関数 $Q : \Sigma^* \to \{0,1\}$ は非可算個存在するという点に注目すればよい．それでは，定理 1.31 から得られるが，濃度に関する議論からは決して得られない情報とは何であるだろうか．それは，停止問題 Halt が「半分，計算可能」であるという点にある．

定義 1.32. 集合 $A \subseteq \mathbb{N}$ が**半計算可能** (*semicomputable*) とは，次を満たすチューリング機械 M が存在することである．

$$(\forall n \in \mathbb{N})\,[n \in A \iff [\![M]\!](n)\downarrow].$$

停止問題 Halt の半計算可能性は定義より自明である．よって，定理 1.31 の結論として，以下を得る．

系 1.33. 計算不可能だが半計算可能な集合が存在する．

豆知識. 実のところ，定義 1.32 の概念について，計算可能性理論において，「半計算可能」と呼ばれることは滅多に無い．理由として，計算可能性理論のもう少し発展的な話題では，半計算可能または**半再帰的** (*semirecursive*) という用語は，これとは全く別の概念に用いられるから，混乱の恐れがあるためであろう．しかし，本ノートでは，そのような発展的話題には触れないので，しばらくはこの用語を用いることとする．

もう少し現代的観点に立つと，定義 1.32 は，**計算可能開集合** (*computable open set*) と言った方がよい代物である．計算論を位相空間論的に理解する方法があり，その立場では，定義 1.32 は，位相空間の開集合に相当するものだからである．これについては，たとえば定義 4.22 で取り扱う．

2　決定問題：解ける問題と解けない問題

定理 1.31 において，停止問題が計算不可能であることを示した．すなわち，チューリング機械 M のコードと入力 n が与えられたときに，M に n を入力した結果が停止するかどうかを計算するアルゴリズムは存在しない．

数学の世界には他にも無数の計算不可能問題が溢れている．これについては，たとえば Poonen によって書かれた “Undecidable problems: a sampler[*2]” というサーヴェイがある．著者の Poonen は，数学の広範な分野（論理学，組合せ論，行列の半群，群論，トポロジー，数論，解析学，力学系，確率論，代数幾何，代数学，ゲーム理論）からそれぞれ幾つかの具体的な計算不可能問題の例をピックアップして紹介している．

しかし，具体的な問題の計算不可能性の証明の多くは，その分野の多少の予備知識を前提とする．本節では，予備知識が殆ど不要で，計算不可能性の証明が短いものだけ幾つか紹介しよう．

2.1　多対一還元

ある問題 Q について，その計算可能な解法は存在しない，ということを示すにはどうすればよいだろうか．そのための 1 つの方法は，その問題 Q が停止問題以上に難しい，と示すことである．定理 1.31 より，停止問題は計算不可能であると知っているから，停止問題以上に難しい問題 Q も計算不可能なはずである．では，どうすれば，ある問題が別の問題より難しいと言えるだろうか．その答えの 1 つは，還元という概念によって与えられる．

定義 2.1. 集合 $A, B \subseteq \mathbb{N}$ に対して，A が B に**多対一還元** (*many-one reducible*) されるとは，ある計算可能関数 $f : \mathbb{N} \to \mathbb{N}$ が存在して，

$$n \in A \iff f(n) \in B$$

を満たすことである．このとき，$A \leqslant_m B$ と書く．

解説. $A \leqslant_m B$ の意味とは，$n \in A$ かどうかを知るためには，$f(n) \in B$ かどうかを知っていればよい，ということである．つまり，問題 A の方が問題 B より簡単である，と考えることができる．これにより，順序 $\leqslant_m$ は，問題の難しさの度合いを表すものと思える．

命題 2.2. $A, B \subseteq \mathbb{N}$ とする．

- $A \leqslant_m B$ であり B が計算可能ならば，A も計算可能である．
- $A \leqslant_m B$ であり B が半計算可能ならば，A も半計算可能である．

[*2] https://arxiv.org/abs/1204.0299 より入手可能.

Proof. $A \leqslant_m B$ ならば，ある計算可能関数 $f : \mathbb{N} \to \mathbb{N}$ が存在して，$n \in A$ と $f(n) \in B$ は同値である．つまり，集合と特性関数を同一視すれば，

$$A(n) = 1 \iff B(f(n)) = 1$$

である．計算可能関数の合成は計算可能関数であるから，B が計算可能ならば $B \circ f$ は計算可能であり，よって，A は計算可能である．

もし B が半計算可能ならば，あるチューリング機械 M が存在して，$n \in B$ であることと $[\![M]\!](n)\downarrow$ であることが同値となる．したがって，

$$n \in A \iff f(n) \in B \iff [\![M]\!](f(n))\downarrow$$

である．計算可能関数の合成は計算可能関数であるから，$[\![M]\!] \circ f$ は計算可能なので，A は半計算可能であることが従う． □

多対一還元の概念を用いると，停止問題が，半計算可能問題の中で最も難しい問題であるということが分かる．

命題 2.3. 集合 $A \subseteq \mathbb{N}$ について，次が成立する．

$$A \text{ は半計算可能である} \iff A \leqslant_m \mathsf{Halt}.$$

Proof. Halt は半計算可能なので，$A \leqslant_m \mathsf{Halt}$ ならば，命題 2.2 より，A は半計算可能である．逆に，A が半計算可能ならば，ある $e \in \mathbb{N}$ について，$n \in A$ であることと $[\![e]\!](n)\downarrow$ であることが同値となる．つまり，$n \in A$ と $(e,n) \in \mathsf{Halt}$ が同値であるから，計算可能関数 $n \mapsto (e,n)$ によって，$A \leqslant_m \mathsf{Halt}$ が保証される． □

この性質を持ってして，停止問題 Halt は Σ_1**-完全** (Σ_1-*complete*) であると呼ばれる．

演習問題 2.4. $\mathsf{Halt}_0 = \{e \in \mathbb{N} : [\![e]\!](0)\downarrow\}$ が計算不可能であることを示せ．

2.2 モノイドの語の問題

それでは，停止問題と多対一還元を利用した計算不可能性証明の例を与えよう．チューリング機械は文字列書換系 (定義 1.11) の一種であるから，以下のように，文字列書換系における到達可能性問題が計算不可能であることを示すことができる．

定理 2.5 (文字列書換系における計算不可能性)**.** 次のような有限文字列書換系 $(A, \to)$ と語 $v \in A^*$ が存在する．与えられた $u \in A^*$ に対して，$u \to^* v$ かどうかを判定する計算可能な方法は存在しない．つまり，

$$\mathsf{Reach}_{\to,v} = \{u \in A^* : u \to^* v\}$$

は計算不可能集合である．

Proof. まず，任意のチューリング機械 M について，$\mathsf{Halt}_M \leqslant_m \mathsf{Reach}_{\rightsquigarrow_M,v}$ となる語 v を持つような文字列書換系 $(A, \rightsquigarrow_M)$ を作りたい．チューリング機械 M について，$(A_M, \longrightarrow_M)$ を定義 1.2 の有限文字列書換系とする．まず思い付くアイデアとしては，計算が受理状態に達した時点での状況を表す文字列を v とすることである．すると，u を初期状況とする計算が停止することと $u \longrightarrow_M^* v$ であることが同値になりそうである．しかし，受理状態の状況 v は出力文字列などの情報も含んでいるので，つまり，計算が受理状態に達した時点での状況を表す文字列 v というものは一種類ではない．この問題を解決するために，受理状態の如何なる状況に到達しても，特定の文字列に必ず収束するように新たな書換規則を付け加える．

アルファベット A_M に新しい記号 $\downarrow$ を加える．$\mathtt{a} \in \Sigma \cup \{\sqcup\}$ と $\mathtt{b} \in \Sigma \cup \{\sqcup, [\}$ について，以下の 3 つの書換規則 $\rightsquigarrow$ を考える．

$$(1)\quad \boxed{q_{\mathrm{end}}}\Rrightarrow \mathtt{a} \rightsquigarrow \boxed{q_{\mathrm{end}}}\Rrightarrow \qquad (2)\quad \boxed{q_{\mathrm{end}}}\Rrightarrow\ ] \rightsquigarrow \downarrow \qquad (3)\quad \mathtt{b}\downarrow \rightsquigarrow \downarrow$$

もしチューリング機械 M が受理状態に達しているとすると，その場合の状況は

$$[\ \sqcup \mathtt{b}_0\mathtt{b}_1 \dots \mathtt{b}_{k-1}\mathtt{b}_k \boxed{q_{\mathrm{end}}}\Rrightarrow \mathtt{b}_{k+1}\mathtt{b}_{k+2}\mathtt{b}_{k+3} \dots \mathtt{b}_t\sqcup\ ]$$

という形になっているはずである．この状況を表す文字列は，規則 $\rightsquigarrow$ によって，以下のように書き換えられる．

$$\begin{aligned}
&\rightsquigarrow [\ \sqcup \mathtt{b}_0\mathtt{b}_1 \dots \mathtt{b}_{k-1}\mathtt{b}_k \boxed{q_{\mathrm{end}}}\Rrightarrow \mathtt{b}_{k+2}\mathtt{b}_{k+3} \dots \mathtt{b}_t\sqcup\ ]\\
&\rightsquigarrow [\ \sqcup \mathtt{b}_0\mathtt{b}_1 \dots \mathtt{b}_{k-1}\mathtt{b}_k \boxed{q_{\mathrm{end}}}\Rrightarrow \mathtt{b}_{k+3} \dots \mathtt{b}_t\sqcup\ ]\\
&\rightsquigarrow^* [\ \sqcup \mathtt{b}_0\mathtt{b}_1 \dots \mathtt{b}_{k-1}\mathtt{b}_k \boxed{q_{\mathrm{end}}}\Rrightarrow \sqcup\ ]\\
&\rightsquigarrow [\ \sqcup \mathtt{b}_0\mathtt{b}_1 \dots \mathtt{b}_{k-1}\mathtt{b}_k \boxed{q_{\mathrm{end}}}\Rrightarrow\ ]\\
&\rightsquigarrow [\ \sqcup \mathtt{b}_0\mathtt{b}_1 \dots \mathtt{b}_{k-1}\mathtt{b}_k\downarrow \rightsquigarrow [\ \sqcup \mathtt{b}_0\mathtt{b}_1 \dots \mathtt{b}_{k-1}\downarrow \rightsquigarrow^* [\ \sqcup \mathtt{b}_0\downarrow\\
&\rightsquigarrow [\ \sqcup\downarrow \rightsquigarrow [\ \downarrow \rightsquigarrow \downarrow
\end{aligned}$$

したがって，受理状態における状況が何であれ，文字列は $\downarrow$ に書き換えられる．書換規則 $\rightsquigarrow_M$ をチューリング機械 M の遷移規則 $\longrightarrow_M$ に上記の規則 $\rightsquigarrow$ を加えたもの，つまり $\rightsquigarrow_M = \longrightarrow_M \cup \rightsquigarrow$ と定義する．すると，入力文字列 $\mathtt{a}_0\mathtt{a}_1 \dots \mathtt{a}_n \in \Sigma^*$ に対して M の計算が停止するならば，次式が成立することが分かる．

$$[\ \sqcup \boxed{q_{\mathrm{init}}}\Rrightarrow \mathtt{a}_0\mathtt{a}_1 \dots \mathtt{a}_n\sqcup\ ] \rightsquigarrow_M^* \downarrow$$

逆に，$[\ \sqcup\ \boxed{q_{\mathrm{init}}}\Rrightarrow \mathtt{a}_0\mathtt{a}_1\dots\mathtt{a}_n\sqcup\ ]$ から開始して，有限回の書換規則 $\rightsquigarrow_M$ の適用で $\downarrow$ に到達したとする．初期文字列に $\downarrow$ は含まれていないから，途中の書換規則の適用で $\downarrow$ が出現したということである．書換規則 $\rightsquigarrow$ の定義を見れば，$\downarrow$ が新しく現れるためには，文字列が $\boxed{q_{\mathrm{end}}}\Rrightarrow$ を含んでいる必要があることが分かる．書換規則 $\rightsquigarrow$ が新しく $\boxed{q_{\mathrm{end}}}\Rrightarrow$ を出現させることはないので，$\boxed{q_{\mathrm{end}}}\Rrightarrow$ の出現は $\rightsquigarrow_M$ によってなされる．しかし，これが意味することは，上の初期文字列を初期状況とする M の計算が受理状態 q_{end} に辿り着くということである．つまり，入力文字列 $\mathtt{a}_0\mathtt{a}_1\dots\mathtt{a}_n\in\Sigma^*$ に対して M の計算が停止する．以上をまとめると，

$$\mathtt{a}_0\mathtt{a}_1\dots\mathtt{a}_n\in\mathsf{Halt}_M\iff[\ \sqcup\ \boxed{q_{\mathrm{init}}}\Rrightarrow\mathtt{a}_0\mathtt{a}_1\dots\mathtt{a}_n\sqcup\ ]\in\mathsf{Reach}_{\rightsquigarrow_M,\downarrow}$$

を得る．よって，計算可能関数 $f:\mathtt{a}_0\mathtt{a}_1\dots\mathtt{a}_n\mapsto[\ \sqcup\ \boxed{q_{\mathrm{init}}}\Rrightarrow\mathtt{a}_0\mathtt{a}_1\dots\mathtt{a}_n\sqcup\ ]$ によって $\mathsf{Halt}_M\leqslant_m\mathsf{Reach}_{\rightsquigarrow_M,\downarrow}$ が保証される．このとき，U を万能チューリング機械とすれば，定理 1.31 より Halt_U は計算不可能であるから，命題 2.2 より $\mathsf{Reach}_{\rightsquigarrow_U,\downarrow}$ は計算不可能である． □

定義 1.23 とその下のパラグラフで説明したように，任意のモノイドは文字列書換系として表示することができる．特に，モノイドの各元は文字列として表され，モノイドの演算は文字列操作によって取り扱うことができる．しかし，問題となるのは，表示を 1 つ固定しても，モノイドの 1 つの要素に対して，それを表す文字列は複数存在し得るという点である．モノイドなどの代数構造を文字列操作として取り扱う場合，どの文字列の組が同じ要素を表すかくらいは判定できてほしい，と考えるのは妥当な要求であるように思える．

しかし，定理 2.5 の応用として分かることは，与えられた 2 つの文字列が，モノイドの同じ要素を表すかどうかについて，アルゴリズム的に判断することは残念ながら不可能である，ということである．これを数学的に正確に述べよう．A 上の与えられた 2 項関係 R に対して，$\equiv_R$ を例 1.19 のように R を含む自由モノイド A^* 上の最小の合同関係として定義する．すると，次のような計算不可能性を得られる．

定理 2.6 (モノイドの語の問題の計算不可能性)**.** 次のような有限表示モノイド $\langle A\mid R\rangle$ と語 $v\in A^*$ が存在する．与えられた $u\in A^*$ に対して，$u\equiv_R v$ かどうかについて，計算可能な判定方法は存在しない．特に，

$$\mathsf{WP}_{\langle A|R\rangle}=\{(u,v)\in A^*\times A^*:u\equiv_R v\}$$

は計算不可能集合である．

Proof. チューリング機械 M について，$(A_M\cup\{\downarrow\},\rightsquigarrow_M)$ を定理 2.5 の有限文字列書換系とする．第 1.2 節で述べたように，有限文字列書換系とモノイドの有限表示は同一であるから，$\langle A\mid\rightsquigarrow_M\rangle$ は有限表示モノイドであると考えられる．$\equiv_M$ を $\rightsquigarrow_M$ を含む最小の合同関係とする．

このとき，u が チューリング機械 M の計算状況を表す語であるならば，$u\equiv_M\downarrow$ であることと $u\rightsquigarrow_M^*\downarrow$ であることが同値であることを示したい．もし，$u\equiv_M\downarrow$ であれば，ある語

$u_0, \ldots u_n \in A^*$ が存在して，たとえば

$$u \leftsquigarrow_M u_0 \leftsquigarrow_M u_1 \rightsquigarrow_M \cdots \leftsquigarrow_M u_{n-1} \leftsquigarrow_M u_n \rightsquigarrow_M \downarrow$$

のようになるはずである．ここで，矢印の向きはもしかすると逆かもしれない．しかし，$\rightsquigarrow_M$ の定義より，$\downarrow \rightsquigarrow_M u$ となる語 $u \in A^*$ は存在しないから，$u_n \rightsquigarrow_M \downarrow$ は確定している．

$B = \{\boxed{q}\Rrightarrow : q \in Q\} \cup \{\downarrow\}$ とする．さらに，W_1 を $A_M \cup \{\downarrow\}$ の語のうち，B の記号を正確に 1 つだけ含むもの全体の集合とする．たとえば，チューリング機械 M の計算状況を表す語は必ず W_1 に含まれる．さて，書換規則 $\rightsquigarrow_M$ の左右を見ると，書換前と書換後の語で，B に属する記号の数は変わらない．特に，$u \equiv_M v$ かつ $u \in W_1$ ならば $v \in W_1$ である．さらに，書換規則 $\rightsquigarrow_M$ をよく見ると，$u \in W_1$ ならば，$u \rightsquigarrow_M v$ なるような語 v は高々 1 つしか存在しないことがわかる．たとえば $\longrightarrow_M$ については，定義 1.3 の直後のパラグラフで言及した通りであり，定理 2.5 で新たに付加した $\rightsquigarrow$ の部分については明らかであろう．

さて，$u \in W_1$ であれば，上のような $u_0, \ldots, u_n$ は全て W_1 に属す．$i \neq j$ のとき，$u_i \neq u_j$ であることは仮定できる．よって，どんな i を見ても，

$$u_{i-1} \leftsquigarrow_M u_i \rightsquigarrow_M u_{i+1}$$

となることは有り得ない．これより，$u_0, \ldots u_n$ の矢印の向きは必ず，

$$u \rightsquigarrow_M u_0 \rightsquigarrow_M u_1 \rightsquigarrow_M \cdots \rightsquigarrow_M u_{n-1} \rightsquigarrow_M u_n \rightsquigarrow_M \downarrow$$

となるはずである．以上より，$u \in W_1$ であれば，$u \rightsquigarrow_M^* \downarrow$ であることと $u \equiv_M \downarrow$ であることが同値であることが示された．

定理 2.5 の証明中の計算可能関数 $f : \mathtt{a_0a_1 \ldots a_n} \mapsto [\ \sqcup\ \boxed{q_{\mathrm{init}}}\Rrightarrow \mathtt{a_0a_1 \ldots a_n}\sqcup\ ]$ について，$\mathtt{a_0a_1 \ldots a_n}$ とすると，必ず $f(\mathtt{a_0a_1 \ldots a_n}) \in W_1$ である．よって，

$$\mathtt{a_0a_1 \ldots a_n} \in \mathsf{Halt}_M \iff f(\mathtt{a_0a_1 \ldots a_n}) \rightsquigarrow_M^* \downarrow \iff f(\mathtt{a_0a_1 \ldots a_n}) \equiv_M \downarrow$$

1 つ目の同値性は，定理 2.5 の証明より成立し，2 つ目の同値性は，W_1 上での $\rightsquigarrow_M^*$ と $\equiv_M$ の同値性による．よって，$\mathsf{Halt}_M \leqslant_m \{u \in A_M \cup \{\downarrow\} : u \equiv_M \downarrow\}$ となる．以上より，U を万能チューリング機械とすれば，定理 1.31 より，Halt_U は計算不可能であるから，命題 2.2 より，$\{u \in A_M \cup \{\downarrow\} : u \equiv_M \downarrow\}$ は計算不可能である． □

定理 2.6 は，モノイド M の具体的な有限表示 $\langle A \mid R\rangle$ に対して，語の問題の計算不可能性を示すものである．しかし，実は，語の問題の計算不可能性は，表示には依存しない．つまり，M のある有限表示 $\langle A \mid R\rangle$ における語の問題が計算不可能であれば，如何なる有限表示 $\langle B \mid S\rangle$ における語の問題も計算不可能である．

命題 2.7. M をモノイドとし，$\langle A \mid R\rangle$ と $\langle B \mid S\rangle$ は共に M の有限表示であるとする．このとき，次が成立する．

$$\mathsf{WP}_{\langle A|R\rangle} \equiv_m \mathsf{WP}_{\langle B|S\rangle}.$$

Proof. ϕ を $\langle A \mid R\rangle$ と $\langle B \mid S\rangle$ の間の同型とする．このとき，各 $a \in A$ について，$\phi(a) \in B^*$ である．集合 A は有限なので，$\phi \restriction A$ は有限関数であるから，計算可能である．A は n 元集合 $\{a_1, \ldots, a_n\}$ であると仮定する．いま，n 個の変数 $\overline{x} = (x_1, \ldots, x_n)$ を持つ項を以下のように帰納的に定義する．各 x_i は項であり，$s(\overline{x})$ と $t(\overline{x})$ が項ならば，$s(\overline{x})$ と $t(\overline{x})$ の結合 $s(\overline{x})t(\overline{x})$ も項である．各 $w \in A^*$ は，ある項 t_w に対して，$w = t_w(a_1, \ldots, a_n)$ という形である．このとき，$\phi(w) = t_w(\phi(a_1), \ldots, \phi(a_n)) \in B^*$ である．明らかに $w \mapsto t_w(\phi(a_1), \ldots, \phi(a_n))$ は計算可能である．よって，

$$u \equiv_R v \iff t_u(\phi(a_1), \ldots, \phi(a_n)) \equiv_S t_v(\phi(a_1), \ldots, \phi(a_n))$$

が成立するから，$\mathsf{WP}_{\langle A|R\rangle} \leqslant_m \mathsf{WP}_{\langle B|S\rangle}$ が導かれる． □

2.3 ポストの対応問題

つづいて，ドミノを思い浮かべよう．ドミノというものをドミノ倒しで知った筆者は，長らく意識したことが無かったのであるが，よく見るとドミノ牌には上下に賽の目のような記号が書かれている．賽の目を書くのは面倒なので，数字で表すこととすると，$\boxed{\begin{smallmatrix}5\\3\end{smallmatrix}}$ のような感じである．

ここでは，一般化されたドミノを考えよう．一般化されたドミノ牌では，上下には賽の目の代わりに語が書かれているとする．たとえば，

$$\boxed{\begin{smallmatrix}\texttt{a}\\\texttt{abc}\end{smallmatrix}}\ \boxed{\begin{smallmatrix}\texttt{b}\\\texttt{bc}\end{smallmatrix}}\ \boxed{\begin{smallmatrix}\texttt{ba}\\\texttt{a}\end{smallmatrix}}\ \boxed{\begin{smallmatrix}\texttt{bca}\\\texttt{ab}\end{smallmatrix}}\ \boxed{\begin{smallmatrix}\texttt{domino}\\\texttt{post}\end{smallmatrix}}$$

のような感じである．このとき，次の問題を考える．

決定問題. 与えられたドミノ牌を上手く配置することによって，上部と下部の語をマッチさせることができるだろうか．ただし，同じドミノ牌は複数回使ってよいし，使わないドミノ牌があってもよい．

たとえば，上の 5 つのドミノ牌であれば，

$$\boxed{\begin{smallmatrix}\texttt{a}\\\texttt{abc}\end{smallmatrix}}\ \boxed{\begin{smallmatrix}\texttt{bca}\\\texttt{ab}\end{smallmatrix}}\ \boxed{\begin{smallmatrix}\texttt{b}\\\texttt{bc}\end{smallmatrix}}\ \boxed{\begin{smallmatrix}\texttt{bca}\\\texttt{ab}\end{smallmatrix}}\ \boxed{\begin{smallmatrix}\texttt{ba}\\\texttt{a}\end{smallmatrix}}$$

と配置すれば，上部の語，下部の語は共に abcabbcaba となり，マッチしている．ここで，ドミノ牌 $\boxed{\begin{smallmatrix}\texttt{bca}\\\texttt{ab}\end{smallmatrix}}$ を 2 回使っており，$\boxed{\begin{smallmatrix}\texttt{domino}\\\texttt{post}\end{smallmatrix}}$ は 1 回も使っていないが，そのようなことは許されている．

もう少し数学的に正確な定式化をしよう．以後 k 未満の自然数の集合を $\mathbf{k} = \{0, 1, \ldots, k-1\}$ と表す．アルファベット A を固定したとき，与えられた有限個のドミノ牌 $(u, v) = \left(\boxed{\begin{smallmatrix}u(i)\\v(i)\end{smallmatrix}} : i < k\right) \in (A^* \times A^*)^k$ について，空でない語 $a_0 a_1 \ldots a_\ell \in \mathbf{k}^*$ で

$$u(a_0)u(a_1)\ldots u(a_\ell) = v(a_0)v(a_1)\ldots v(a_\ell)$$

となるものが存在するだろうか．これを判定せよ，という問題が**ポストの対応問題** (*Post's correspondence problem*) である．

定理 2.8. 十分大きな $k \in \mathbb{N}$ について，k 個のドミノ牌に関するポストの対応問題 PCP_k は計算不可能である．つまり，次の集合は計算不可能である．

$$\mathsf{PCP}_k = \{(u,v) \in (A^* \times A^*)^k : (\exists a_0 a_1 \dots a_\ell \in \mathbf{k}^* \backslash \{\varepsilon\}) \\ u(a_0)u(a_1)\dots u(a_\ell) = v(a_0)v(a_1)\dots v(a_\ell)\}$$

Proof. 定理 2.5 より，有限文字列書換系 $(A, \to)$ で，$\mathsf{Reach}_\to = \{(u,v) \in A^* \times A^* : u \to^* v\}$ が計算不可能であるものが存在する．$\mathsf{Reach}_\to \leqslant_m \mathsf{PCP}_k$ を示そう．ここで，$k = 2|A| + |\to| + 2$ とする．

このために，アルファベット $A \cup \{\overline{a} : a \in A\} \cup \{\#\}$ を固定する．k 個のドミノ牌は，以下によって与えられる．与えられた語 $u, v \in A^*$ に対して，ドミノ牌 $\left[\begin{smallmatrix}\#u\\\#\end{smallmatrix}\right]$ と $\left[\begin{smallmatrix}\#\\v\#\end{smallmatrix}\right]$ を用意する．また，各 $a \in A$ について，ドミノ牌 $\left[\begin{smallmatrix}a\\\overline{a}\end{smallmatrix}\right]$ と $\left[\begin{smallmatrix}\overline{a}\\a\end{smallmatrix}\right]$ を用意する．さらに，各書換規則 $r \to s$ について，$\left[\begin{smallmatrix}\overline{s}\\r\end{smallmatrix}\right]$ を用意する．ここで，$s = s_0 s_1 \dots s_n \in A^*$ とすると，$\overline{s} = \overline{s_0 s_1} \dots \overline{s_n}$ である．

まず，$u \to^* v$ ならば，$u \to w_1 \to w_2 \to \cdots \to w_i \to v$ となる語の列 $(w_j)_{j \leqslant i}$ が存在する．よって，ドミノ牌を

$$\left[\begin{smallmatrix}\#u\\\#\end{smallmatrix}\right] \left[\begin{smallmatrix}\overline{w_1}\\u\end{smallmatrix}\right] \left[\begin{smallmatrix}w_1\\\overline{w_1}\end{smallmatrix}\right] \left[\begin{smallmatrix}\overline{w_2}\\w_1\end{smallmatrix}\right] \left[\begin{smallmatrix}w_2\\\overline{w_2}\end{smallmatrix}\right] \cdots \left[\begin{smallmatrix}w_i\\\overline{w_i}\end{smallmatrix}\right] \left[\begin{smallmatrix}\overline{v}\\w_i\end{smallmatrix}\right] \left[\begin{smallmatrix}v\\\overline{v}\end{smallmatrix}\right] \left[\begin{smallmatrix}\#\\v\#\end{smallmatrix}\right]$$

と配置すれば上部と下部をマッチさせられる．

逆に，上で用意したドミノ牌を上手く配置すると，上部と下部をマッチさせられると仮定する．左端が一致するドミノ牌は $\left[\begin{smallmatrix}\#u\\\#\end{smallmatrix}\right]$ しかないので，これが最初に配置される．このとき，上部にはバーなし文字の列 u が書かれているため，上下がマッチしているということは，下部にバーなし文字が書かれたドミノ牌が次に配置されているはずである．したがって，次に配置されている可能性のあるドミノは，$\left[\begin{smallmatrix}\overline{a}\\a\end{smallmatrix}\right]$ か $\left[\begin{smallmatrix}\overline{s}\\r\end{smallmatrix}\right]$ の形のものである．その後，下部の文字列が u になるまで，この形のドミノ牌が配置され続けているはずである．

下部の文字列で u が作られた段階で，上部に文字列 $\overline{w_1}$ が出来ていると仮定しよう．つまり，ドミノ牌を連結すれば $\left[\begin{smallmatrix}\#u\overline{w_1}\\\#u\end{smallmatrix}\right]$ となっている．この $\left[\begin{smallmatrix}\overline{w_1}\\u\end{smallmatrix}\right]$ の部分は $\left[\begin{smallmatrix}\overline{a}\\a\end{smallmatrix}\right]$ か $\left[\begin{smallmatrix}\overline{s}\\r\end{smallmatrix}\right]$ の形のドミノ牌を連結して作られたものである．このとき，$a \to^* a$ かつ $r \to^* s$ であるから，$u \to^* w_1$ が成立する．

いま，上部にはバー付き文字の列 $\overline{w_1}$ が余分に現れているので，上下がマッチしているということは，下部にバー付き文字が書かれたドミノ牌が次に配置されているはずである．したがって，続いて配置され得るドミノは，$\left[\begin{smallmatrix}a\\\overline{a}\end{smallmatrix}\right]$ の形のものしか有り得ない．先ほどと同様に，$\overline{w_1}$ を下部に配置し終えるまで，この形のドミノが配置され続ける．その結果，ドミノを連結すると $\left[\begin{smallmatrix}\#u\overline{w_1}w_1\\\#u\overline{w_1}\end{smallmatrix}\right]$ という形になる．そうすると，またバーなし文字の列 w_1 が余分に現れるから，次は下部にバーなし文字が書かれたドミノ牌が次に配置される．

この手続きを繰り返していくと，$w_1 \to^* w_2 \to^* \cdots \to^* w_i$ という列を次々に得て，ドミノを連結すると $\left[\begin{smallmatrix}\#u\overline{w_1}w_1\overline{w_2}w_2\dots\overline{w_i}w_i\\\#u\overline{w_1}w_1\overline{w_2}w_2\dots\overline{w_i}\end{smallmatrix}\right]$ という形になる．さて，右端が一致するドミノは $\left[\begin{smallmatrix}\#\\v\#\end{smallmatrix}\right]$ しかないの

で，上部と下部がマッチするためには，このドミノが最後に配置される必要がある．つまり，上部と下部がマッチする状況というのは，$\boxed{\begin{matrix}\#u\overline{w_1}w_1\overline{w_2}w_2\ldots\overline{w_i}w_i\#\\ \#u\overline{w_1}w_1\overline{w_2}w_2\ldots\overline{w_i}v\#\end{matrix}}$ という場合しか有り得ない．以上の議論より，ある語の列 $(w_j)_{j\leqslant i}$ が存在して，

$$u \to^* w_1 \to^* w_2 \to^* \cdots \to^* w_i = v$$

となるから，$u \to^* v$ となることが示された．

与えられた (u, v) に対して，上のような k 種類のドミノの列 D を作る手続きは計算可能であり，また，$u \to^* v$ であることと $D \in \mathsf{PCP}_k$ であることは同値だと示したから，$\mathsf{Reach}_{\to} \leqslant_m \mathsf{PCP}_k$ を得る．$\mathsf{Reach}_{\to}$ は計算不可能であったから，命題 2.2 より，PCP_k は計算不可能である． □

演習問題 2.9. 定理 2.8 において，アルファベット A は 2 つの記号からなるものでよいことを示せ．

2.4 行列のモータリティ問題

つづいて，行列に関する決定問題を考えよう．有限個の正方行列 $M_0, M_1, \ldots, M_k$ が与えられたとき，これらの行列を用いた有限積で零行列を作れるかどうかを判定したい．ここで，ポストの対応問題の場合と同様に，同じ行列を複数回使ってもよいし，使わない行列があってもよいものとする．簡単な例を挙げると，たとえば

$$\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}, \quad \begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix}, \quad \begin{pmatrix} 2 & 2 \\ 3 & 4 \end{pmatrix}$$

という 3 つの行列が与えられているとする．すると，以下のように零行列を作ることができる．

$$\begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}\begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} = \begin{pmatrix} -1 & 1 \\ -1 & 1 \end{pmatrix}^2 = O$$

一方，たとえば次のような行列が与えられているとする．

$$\begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}, \quad \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}, \quad \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}.$$

これらの行列を用いた積では決して零行列を作ることはできない．なぜなら，上の 3 つの行列の行列式はいずれも 0 でないが，行列式が 0 でない行列をいくら掛けても，決して行列式が 0 になることはないからである．

行列のモータリティの問題 (*matrix mortality problem*) とは，$d \times d$ 整数行列の有限集合 $\mathcal{M} = (M_0, \ldots, M_{m-1}) \in M_d(\mathbb{Z})^m$ が与えられたとき，$\mathcal{M}$ の中の行列の有限個の積で零行列が作れるかどうかを判定する問題である．より正確には，$\mathbf{m} = \{0, \ldots, m-1\}$ 上の語 $w = a_0 a_1 \ldots a_s$ に対して，

$$\mathcal{M}_w = M_{a_0} M_{a_1} \cdots M_{a_s}$$

と書くとすれば，$\mathcal{M}_w = O$ となる語 $w \in \mathbf{m}^*$ が存在するかどうかを判定せよ，という問題である．ここで O は零行列を表すものとする．

より数学的に言い換えれば，与えられた有限集合 $S \subseteq M_d(\mathbb{Z})$ に対して，S が生成する乗法半群が零行列を含むか否かを判定する問題である．

定理 2.10. 十分大きな $m \in \mathbb{N}$ について，m 個の 3×3 整数行列のモータリティ問題は計算不可能である．つまり，以下の集合は計算不可能である．

$$\mathsf{MM}_m = \{\mathcal{M} \in M_3(\mathbb{Z})^m : (\exists w \in \mathbf{m}^*)\ \mathcal{M}_w = O\}.$$

Proof. 4 進アルファベット $\{0,1,2,3\}$ 上の語 u が与えられたとき，4 進展開が u によって表される自然数を $(u)_4$ と書く．語 $u, v \in \{1,2,3\}^*$ が与えられたとき，行列 $M_{u,v} \in M_3(\mathbb{Z})$ を以下によって定義する．

$$M_{u,v} = \begin{pmatrix} 4^{|u|} & 0 & 0 \\ 0 & 4^{|v|} & 0 \\ (u)_4 & (v)_4 & 1 \end{pmatrix}$$

この $(u,v) \mapsto M_{u,v}$ は単射モノイド準同型である．まず，単射性は明らかである．準同型性，つまり，$M_{u,v}M_{s,t} = M_{us,vt}$ であることは以下によって示される．

$$M_{u,v}M_{s,t} = \begin{pmatrix} 4^{|u|}\cdot 4^{|s|} & 0 & 0 \\ 0 & 4^{|v|}\cdot 4^{|t|} & 0 \\ 4^{|s|}(u)_4 + (s)_4 & 4^{|t|}(v)_4 + (t)_4 & 1 \end{pmatrix} = \begin{pmatrix} 4^{|us|} & 0 & 0 \\ 0 & 4^{|vt|} & 0 \\ (us)_4 & (vt)_4 & 1 \end{pmatrix} = M_{us,vt}$$

2 つめの等号については，$|u|+|s| = |us|$ であり，数の 4 進展開を考えれば，$4^{|s|}(u)_4 + (s)_4 = (us)_4$ であることから従う．つづいて，次の行列 B を考える．

$$B = \begin{pmatrix} 1 & 0 & 1 \\ -1 & 0 & -1 \\ 0 & 0 & 0 \end{pmatrix}$$

このとき，$B^2 = B$ であることは容易に確認できる．また，

$$B\begin{pmatrix} a & 0 & 0 \\ 0 & b & 0 \\ c & d & 1 \end{pmatrix}B = \begin{pmatrix} a+b & d & 1 \\ -a-b & -d & -1 \\ 0 & 0 & 0 \end{pmatrix}B = \begin{pmatrix} a+b-d & 0 & a+b-d \\ -a-b+d & 0 & -a-b+d \\ 0 & 0 & 0 \end{pmatrix}$$

であるから，特に $BM_{u,v}B = (4^{|u|} + (u)_4 - (w)_4)B = ((1u)_4 - (w)_4)B$ となる．このとき，

$$BM_{u,v}B = O \iff (1u)_4 = (v)_4 \iff 1u = v.$$

これらの性質を利用して，$\mathsf{PCP}_k \leqslant_m \mathsf{MM}_{2k+1}$ を示す．ここで，定理 2.8 より，k はポストの対応問題 PCP_k が計算不可能であるような十分大きな k とする．ドミノに用いられるアルファベットは $\{2,3\}$ であるとする．与えられた k 個のドミノ $(u,v) = \left(\left[\begin{smallmatrix} u(i) \\ v(i) \end{smallmatrix}\right] : i < k\right)$ に対して，$2k+1$ 個の行列 $(M_i)_{i<k}$, $(M'_i)_{i<k}$ および B を考える．ここで，

$$M_i := M_{u(i),v(i)}, \quad M'_i := M_{u(i),1v(i)}.$$

まず，$(u,v) \in \mathsf{PCP}_k$ を仮定する．このとき，ある語 $w = a_0a_1\dots a_n \in \mathbf{k}^*$ でドミノの対応が作れる．これは $u(a_0)u(a_1)\dots u(a_n) = v(a_0)v(a_1)\dots v(a_n)$ を意味するので，特に $1u(a_0)u(a_1)\dots u(a_n) = 1v(a_0)v(a_1)\dots v(a_n)$ である．上で示したように，これは

$$BM_{u(a_0)\dots u(a_n),1v(a_0)\dots v(a_n)}B = BM'_{u(a_0),v(a_0)}M_{u(a_1),v(a_1)}\cdots M_{u(a_n),v(a_n)}B = O$$

を意味するので，つまり，上の $2k+1$ 個の行列の組合せで零行列を作ることができた．よって，$((M_i)_{i<k},(M'_i)_{i<k},B) \in \mathsf{MM}_{2k+1}$ が示された．

逆に上の $2k+1$ 個の行列のある組合せで零行列を作れると仮定しよう．この組合せは，$B^2 = B$ かつ $M_{u,v}M_{s,t} = M_{us,vt}$ であることを利用すれば，

$$BM_{s_0,t_0}BM_{s_1,t_1}BM_{s_2,t_2}B\dots BM_{s_\ell,t_\ell}B$$

という形であると仮定できる．しかし，$BM_{s,t}B$ は B のスカラー倍であるから，ある i について $1s_i = t_i$ であるときのみしか，上の組合せは零行列になることは有り得ない．上での $2k+1$ 個の行列の選び方より，s_i は $(u(j))_{j<k}$ の組合せで作られているので，$s_i \in \{2,3^*\}$ である．一方，$1s_i = t_i$ より，t_i の最初の文字は 1 である．よって，

$$M_{s_i,t_i} = M'_{r_0}M_{r_1}\cdots M_{r_j}$$

という形になっている．これより，

$$1u(r_0)u(r_1)\dots u(r_j) = 1s_i = t_i = 1v(r_0)v(r_1)\dots v(r_j)$$

であるから，語 $r_0r_1\dots r_j$ はドミノの対応を与える．

以上より，$(u,v) \in \mathsf{PCP}_k$ であることと $((M_i)_{i<k},(M'_i)_{i<k},B) \in \mathsf{MM}_{2k+1}$ であることの同値性が示された．還元 $(u,v) \mapsto ((M_i)_{i<k},(M'_i)_{i<k},B) \in \mathsf{MM}_{2k+1}$ は明らかに計算可能であるから，これより，$\mathsf{PCP}_k \leqslant_m \mathsf{MM}_{2k+1}$ を得る．よって，ポストの対応問題 PCP_k の計算不可能性と命題 2.2 より，行列のモータリティ問題 MM_{2k+1} も計算不可能であることが示された． □

2.5 その他の決定問題 *

計算不可能な問題は，数学の様々な分野で現れる．そのうちの幾つかに触れておこう．あらかじめ述べておくと，この節の内容には証明を与えない．

■語の問題とトポロジーにおける決定問題: 定理 2.6 では，モノイドの語の問題の計算不可能性を示した．これは，半群の語の問題の計算不可能性の証明にもなっている．予想は付くと思うが，群の語の問題もまた計算不可能である．しかし，その証明は意外にも非自明である．そんなに難しいというほどでもないが，本稿の主題から外れる内容となるので，証明は省略する．さて，群の語の問題は，もともとトポロジーの文脈で興味を持たれてきた．まず，有限表示群 $\langle A \mid R\rangle$ が与えられたとき，基本群が $\langle A \mid R\rangle$ と同型であるような 2 次元単体複体 $C_{A,R}$ を作ることができる．

$$\pi_1(C_{A,R}) \simeq \langle A \mid R\rangle.$$

これを利用すると，群の語の問題はトポロジーにおける次の問題に翻訳される．

定理 2.11. 与えられた 2 次元単体複体が単連結かどうかを判定する計算可能な方法は存在しない．

このような方針で，トポロジーにおいて，大体 1950 年代末頃から，たとえば次のような問題が計算不可能であることが知られるようになった．

- (マルコフ) $d \geqslant 4$ について，与えられた 2 つの d 次元多様体が同相かどうかを判定する計算可能な方法は存在しない．
- (ノヴィコフ) $d \geqslant 5$ とする．任意の d 次元多様体 M について，与えられた d 次元多様体が P と微分同相かどうかを判定する計算可能な方法は存在しない．

さて，ここまで語の問題が計算不可能であるような群について議論してきた．しかし，世の中には語の問題が計算可能であるような群もかなり多く，応用上はそのような群に注意を向けた方が生産的かもしれない．計算機科学の観点からは，**オートマトン群** (*automatic group*) と呼ばれるタイプの群が重要である．たとえば，有限生成のコセクター群やブレイド群（組み紐群）などがオートマトン群の代表例である．オートマトン群であれば，語の問題は計算可能となることが知られている．

■ヒルベルトの第 10 問題: 群の語から離れると，数学史において最も重要な決定問題は，**ヒルベルトの第 10 問題** (*Hilbert's tenth problem*) に関するものであろう．1900 年の国際数学者会議でヒルベルトが提示した問題は，ヒルベルトの 23 の問題として知られる．その中の第 10 の問題が，与えられた整係数ディオファントス方程式が整数解を持つかどうかを決定するアルゴリズムを見つけよ，というものであった．つまり，有限個の未知数を持つ整数係数多項式 $P(x_1, x_2, \ldots, x_n) = 0$ が与えられたとき，これが解を持つかどうか判定する計算可能な方法は存在するかどうか，ということを知りたい．この問題は 1970 年になってようやく，マチャセビッチによって解決が与えられた．理論的には次の概念が重要である．

定義 2.12. 集合 $A \subseteq \mathbb{N}$ が**ディオファントス的** (*Diophantin*) であるとは，ある整数係数多項式 P が存在して，次の性質を満たすことである．

$$n \in A \iff (\exists x_1, x_2, \ldots, x_k \in \mathbb{N})\ P(n, x_1, x_2, \ldots, x_k) = 0.$$

以下の MRDP 定理が，ヒルベルトの第 10 問題の解決の鍵である．ここで MRDP は，マチャセビッチ (Yuri Matiyasevich), ロビンソン (Julia Robinson), デーヴィス (Martin Davis), パトナム (Hilary Putnam) のファミリーネームの頭文字を並べたものである．

定理 2.13 (MRDP 定理)**.** 集合 $A \subseteq \mathbb{N}$ が半計算可能であることとディオファントス的であることは同値である.

系 1.33 における停止問題をはじめとして，今や我々は沢山の計算不可能な半計算可能集合を知っている．したがって，MRDP 定理 2.13 より，計算不可能なディオファントス集合が存在する．つまり，うまく整数係数多項式 P を作ると，与えられた n について，$P(n, x_1, x_2, \ldots, x_k) = 0$ が整数解を持つかどうかを判定する計算可能な方法が存在しないことが分かる．このようにして，ヒルベルトの第 10 問題の否定的解決が与えられた.

豆知識. 数学の定理がどれくらい弱い公理系で証明可能かを決定することは，しばしば重要な応用を持つ．たとえば，もしペアノ算術を弱めた $I\Delta_0 + \Omega_1$ という公理系で MRDP 定理 2.13 を証明できることを示すことができるならば，NP=co-NP が導かれることが知られている．(……といっても計算理論への数理論理学によるアプローチというものは，計算理論の最初期の時代のポピュラーな手法のひとつであり，古い時代に大勢の手によって限界までやり尽くされている．今の時代に数理論理学的アプローチで計算量理論の大未解決問題を解こうと試みるのはあまりオススメしないかもしれない.)

3 部分組合せ代数

3.1 ストリーム計算と神託機械

ここまでに取り扱った計算モデルは，計算を終了すると計算を「停止」していた．しかし，我々の世代の慣れ親しんだコンピュータが「停止」するのは，電源を切るかフリーズしたときである．数学的には，フリーズというのは，むしろ「停止しない」ことに相当するものであり，電源が入って正常に動いている状態は，「停止」でも「非停止」でもないと言えそうである．このように「停止」も「非停止」も起きずに，コンピュータとその利用者のやり取りが延々続く，あるいはネットワークなどを介したコンピュータ間のやり取りが延々続く計算はどのようにモデル化されるだろうか.

計算可能性理論の次のステップに進むにあたって，まず重要となる概念は，**ストリーム** (*stream*) というデータ型に関する計算である．ストリームとは，次々に与えられるデータの奔流である．数学的には,

$$\mathtt{a}_0\mathtt{a}_1\mathtt{a}_2\mathtt{a}_3 \ldots \mathtt{a}_n\mathtt{a}_{n+1} \ldots$$

という無限に続く文字列と同一視できるが，実際には，時間経過に従って徐々に $\mathtt{a}_0, \mathtt{a}_1, \ldots$ というデータを流し込んでくるものと思う方が都合が良い．つまり，各時刻では，ストリームは，まだ $\mathtt{a}_0\mathtt{a}_1 \ldots \mathtt{a}_k$ というような有限データしか配信していない.

ストリームは基本的には外部情報であるが，我々はストリーム型のデータを利用して計算を行うことが可能である．もう少し正確に議論するために，再びチューリング機械のようなモデルを考えよう．外部からのストリームを流し込むための専用テープを**神託テープ** (*oracle tape*) と呼ぶ．このテープは読込専用で，我々は書込不可能である．これを神託テープと呼ぶよりも，単純に，スト

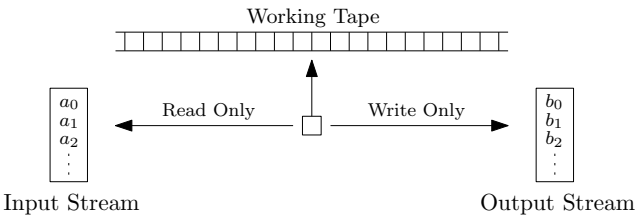


図 1　ストリーム上の関数の計算モデル

リーム型のデータの入力テープだと考えた方が都合がよいことも多い．

そうすると，我々もストリーム型のデータの出力を行いたい．この場合，我々はチューリング機械の計算を停止させずに，延々と稼働させつづけ，文字列を

$$\mathtt{b_0 b_1 b_2 b_3 \ldots b_n b_{n+1} \ldots}$$

というように，時間経過に沿って，次々に出力させていくこととなる．このためには，ストリーム型のデータの出力テープがあると便利である．ただし，一度配信したデータは，別の人物が既に受信しているかもしれないから，もう無かったことにはできない．この状況は，ストリーム出力テープは書込専用だが上書き不可能である，という特性によって定式化できる．

このストリーム入出力テープに加え，読込，書込，上書きが可能な**作業用テープ** (*working tape*) を合わせた計算モデルを考えよう（図 1）．これは，通常，**神託チューリング機械** (*oracle Turing machine*) と呼ばれるものに相当する．さて，この計算モデルにおいて，どのようなストリーム上の関数が計算可能であるだろうか．

この計算モデルを直接取り扱ってもよいが，ストリーム計算のモデルは，非常に単純明快な数学的記述を持つ．これを説明するために，集合 $\mathbb{A}$ を固定する．ただし，$\mathbb{A}$ として，たとえば $\{0,1\}$, Σ, $\mathbb{N}$, Σ^* などを想定している．以後，$\sigma \in \mathbb{A}^*$ が $\tau \in \mathbb{A}^*$ の**始切片** (*initial segment*) であるとき，$\sigma \preceq \tau$ と書く．つまり，$\sigma \preceq \tau$ であるとは，ある $\eta \in \mathbb{A}^*$ について，$\sigma\eta = \tau$ となることである．

定義 3.1. 部分関数 $\varphi :\subseteq \mathbb{A}^* \to \mathbb{A}^*$ が**単調** (*monotone*) とは，以下の条件を満たすことである．

$$(\forall \sigma, \tau \in \mathrm{dom}(\varphi))\ \sigma \preceq \tau \implies \varphi(\sigma) \preceq \varphi(\tau).$$

部分単調関数 $\varphi :\subseteq \mathbb{A}^* \to \mathbb{A}^*$ が与えられたとき，部分関数 $\widehat{\varphi} :\subseteq \mathbb{A}^{\mathbb{N}} \to \mathbb{A}^{\mathbb{N}}$ を次によって定義する．

$$\widehat{\varphi}(X)(n) = m \iff (\exists \sigma \prec X)\ \varphi(\sigma)(n) = m.$$

つまり，集合としては $\widehat{\varphi}(X) = \bigcup_{\sigma \prec X} \varphi(\sigma)$ によって定義する．部分関数 $\Phi :\subseteq \mathbb{A}^{\mathbb{N}} \to \mathbb{A}^{\mathbb{N}}$ が**連続** (*continuous*) とは，ある部分単調関数 φ に対して，$\Phi = \widehat{\varphi}$ が成り立つことである．

豆知識. 集合 $\mathbb{A}$ に離散位相を入れ，$\mathbb{A}^{\mathbb{N}}$ にはその積位相を入れる．このとき，$\Phi :\subseteq \mathbb{A}^{\mathbb{N}} \to \mathbb{A}^{\mathbb{N}}$ が上の意味で連続であることと，位相空間論の意味で連続であることは同値である．

注意. 定義 3.1 の単調関数は常に全域関数であると仮定してよい．なぜなら，部分単調関数 $\varphi :\subseteq \mathbb{A}^* \to \mathbb{A}^*$ が与えられたとき，$\psi(\sigma) = \bigcup\{\varphi(\tau) : \tau \preceq \sigma \text{ and } \varphi(\tau)\downarrow\}$ で定義すれば ψ は全域になるが，φ と ψ が定義する連続関数は同一である．

つまり，連続関数とは，有限文字列上の単調関数 $\varphi :\subseteq \mathbb{A}^* \to \mathbb{A}^*$ によって制御されるストリーム上の関数のことである．ストリームの計算処理を行う機械（神託チューリング機械）は，この有限文字列上の単調関数の部分の計算を行うのみであり，動作としては通常のチューリング機械と全く等しい．ストリーム上の関数が計算可能とは，このような機械によって計算されることである．数学的には，以下のように定式化される．

定義 3.2. 部分関数 $\Phi :\subseteq \mathbb{A}^{\mathbb{N}} \to \mathbb{A}^{\mathbb{N}}$ が**計算可能** (*computable*) または**計算可能連続** (*computably continuous*) とは，ある部分計算可能単調関数 $\varphi :\subseteq \mathbb{A}^* \to \mathbb{A}^*$ に対して，$\Phi = \widehat{\varphi}$ が成り立つことである．

注意. 定義 3.2 の単調関数も常に全域関数であると仮定できる．ただし，定義 3.1 の後の注意のような全域関数 ψ は計算可能とは限らないので，以下の修正が必要である．部分計算可能単調関数 $\varphi :\subseteq \mathbb{A}^* \to \mathbb{A}^*$ が与えられたとき，$\eta(\sigma) = \bigcup\{\varphi(\tau) : \tau \preceq \sigma \text{ and } \varphi(\tau)[|\sigma|]\downarrow\}$ で定義する．ここで $\varphi(\tau)[|\sigma|]\downarrow$ は，$\varphi(\tau)$ を計算するチューリング機械の動作が高々 $|\sigma|$ ステップで停止することを意味する．このように定義すれば，η は全域計算可能単調関数であり，$\widehat{\eta} = \widehat{\varphi}$ となる．

■神託としてのストリーム: 最初に述べたように，ストリーム計算は，現実のコンピュータのような延々やり取りの続く計算モデルを表すものと考えることもできる．しかし，歴史上は，これは神託を用いた計算という，超越的な計算を表すものとして導入された．

$\mathbb{A} = \mathbb{N}$ の場合を考えよう．入力ストリーム $g : \mathbb{N} \to \mathbb{N}$ は，何らかの**神託** (*oracle*) として得られたとしよう．すると，出力ストリーム $f = \Phi(g) : \mathbb{N} \to \mathbb{N}$ は自然数上の関数である．この関数 $f : \mathbb{N} \to \mathbb{N}$ 自体は計算不可能かもしれないが，$g : \mathbb{N} \to \mathbb{N}$ の情報を使うことによって計算できた，ということである．この状況を f は g**-相対的に計算可能** (g-*relatively computable*) または単に g**-計算可能** (g-*computable*) と呼ぶ．

定義 3.3. 関数 $f, g : \mathbb{N} \to \mathbb{N}$ が与えられたとき，f が g に**チューリング還元** (*Turing reducible*) されるとは，f が g-相対的計算可能である，すなわち $f = \Phi(g)$ なる部分計算可能関数 $\Phi :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ が存在することである．このとき，$f \leqslant_T g$ と書く．

神託チューリング機械を疑似プログラミング言語として表現することもできる．これは疑似プログラミング言語 `TURING` に「変数 `y` に神託テープの第 n 番目のセルに書かれた文字を入力せよ」という命令 `y:=Oracle(n)` を新たに加えたものである．これ以外の変更は一切ない．神託チューリング機械のプログラム自体は現実的に記述することができる．単純に神託テープの内容がブラック

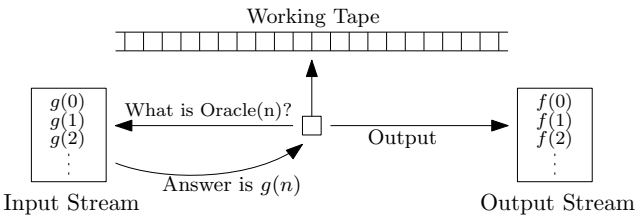


図 2　チューリング還元 $f \leqslant_T g$ の動作

ボックスなだけである．

演習問題 3.4. チューリング還元可能性関係 $\leqslant_T$ が $\mathbb{N}^{\mathbb{N}}$ 上の**前順序** (*preorder*) であることを示せ．つまり，$\leqslant_T$ が反射的 (reflexive) かつ推移的 (transitive) であることを証明せよ．

チューリング還元の簡単な具体例として，以下のようなものを証明してみよう．

命題 3.5. どんな計算可能関数よりも急増大する関数 $f \leqslant_T \mathsf{Halt}$ が存在する．つまり，ある Halt-計算可能関数 $f : \mathbb{N} \to \mathbb{N}$ が存在して，任意の計算可能関数 $g : \mathbb{N} \to \mathbb{N}$ に対して，次が成立する．

$$(\exists s \in \mathbb{N})(\forall n \geqslant s)\ g(n) < f(n).$$

Proof. 関数 $f : \mathbb{N} \to \mathbb{N}$ を次のように定義する．

$$f(n) = 1 + \max\{[\![e]\!](n) : e \leqslant n \text{ and } [\![e]\!](n)\downarrow\}.$$

ここで $\max \varnothing = 0$ とする．入力 n に対して，各 $e \leqslant n$ について $\mathtt{Oracle}(\mathtt{e},\mathtt{n}) = \mathtt{1}$ かどうかを尋ね，そうであるような全ての e について，$[\![e]\!][n]$ をシミュレートし，これら全ての計算が停止するのを待ち，その計算結果の中での最大値 $+1$ を出力する．神託テープに記述されているものが Halt であれば，この手続きは正しく f を計算する．よって，f は Halt-計算可能である．

また，$g : \mathbb{N} \to \mathbb{N}$ が計算可能であれば，ある e について $g = [\![e]\!]$ であるから，f の定義より，任意の $n \geqslant e$ について，$f(n) > g(n)$ となる． □

命題 3.5 の性質を持つ関数の具体例として，**ビジービーバー関数** (*Busy Beaver function*) と呼ばれるものが知られている．

神託，つまり入力ストリームが g である場合，コード e のチューリング機械によって計算される関数（あるいはストリーム）を $[\![e]\!]^g$ によって表す．たとえば，単調関数 φ がコード e のチューリング機械で計算されているとき，定義 3.1 のようにして $\varphi = [\![e]\!]$ から作られる関数 $\widehat{\varphi} = \widehat{[\![e]\!]}$ を考える．このとき，$\widehat{[\![e]\!]}$ に g を入力した結果 $\widehat{[\![e]\!]}(g)$ を $[\![e]\!]^g$ と表すということである．この $[\![e]\!]^g$ をコード e の g-相対的計算可能関数と思うことができる．すると，たとえば，**g-相対的停止問題** (*halting problem relative to* g) などを次のように定義できる．

$$\mathsf{Halt}^g = \{(e,n) : [\![e]\!]^g(n)\downarrow\}.$$

以下，$f <_T g$ によって，$f \leqslant_T g$ かつ $g \not\leqslant_T f$ であることを表す．

定理 3.6. 任意の $g : \mathbb{N} \to \mathbb{N}$ に対して，$g <_T \mathsf{Halt}^g$ が成り立つ．

Proof. 定理 1.31 の停止問題の計算不可能性証明を相対化すればよい． □

以後，Halt^g のことを g の**チューリング・ジャンプ** (*Turing jump*) と呼び，g' と書く．たとえば，$\varnothing''$ は停止問題に相対的な停止問題 $\mathsf{Halt}^{\mathsf{Halt}}$ を表す．定理 3.6 より，以下のようなチューリング順序関係 $\leqslant_T$ の無限増大列を得られる．

$$\varnothing <_T \mathsf{Halt} \equiv_T \varnothing' <_T \varnothing'' <_T \varnothing''' <_T \cdots.$$

最後に，チューリング還元と他の還元を比較しよう．定義 2.1 において，多対一還元可能性の概念を定義した．さて，多対一還元はチューリング還元にいくつかの制約を加えたものと思える．まず，次が成立することは容易に分かる．

$$A \leqslant_m B \implies A \leqslant_T B.$$

しかし，多対一還元では，各入力 n に対して，神託 B の一箇所 $f(n)$ の部分にしか質問をすることができない．つまり，$f(n)$ の値 k を計算するプログラムを記述した後に，命令 $\mathtt{y} := \mathtt{Oracle}(\mathtt{k})$ が一度だけ現れる．さらに，その解答 y がそのまま出力となる．一方，チューリング還元の場合は，プログラム中に $\mathtt{y} := \mathtt{Oracle}(\mathtt{n})$ の形の命令が何度も現れてよいし，クエリの解答をそのまま出力とする必要はない．実際，$A \leqslant_T B$ であっても $A \leqslant_m B$ とはならないことは，後に，系 3.15 として証明する．

3.2 計算可枚挙集合と枚挙還元

自然数の部分集合 $A \subseteq \mathbb{N}$ の枚挙をストリームの一種として考えることができる．つまり，枚挙とは，以下のように自然数を並べていくストリームである．

$$2, 3, 7, \bullet, 11, 5, 7, 13, \bullet, 19, 17, \bullet, \bullet, 23, 3, 29, \ldots$$

ここで，記号 $\bullet$ は何も並べないことを意味する．このストリームは素数の集合の枚挙のつもりである．ここで注意することは，枚挙を考える際，並べる順番は気にしないし，同じものを何度も並べてもよいとする．したがって，たとえば素数の集合を枚挙するストリームは無限パターン存在する．

数学的な定式化を与えれば，**枚挙** (*enumeration*) というものは，関数 $p : \mathbb{N} \to \mathbf{1} + \mathbb{N}$ である．ここで $\mathbf{1} = \{\bullet\}$ であり，$\mathbf{1} + \mathbb{N}$ は $\mathbf{1}$ と $\mathbb{N}$ の和集合を表しているが，和集合の記号 $\cup$ を使わなかったのには理由があり，それは後の節で明らかになる．しかし，今回のケースでは単に

$$\mathbf{1} + \mathbb{N} = \mathbf{1} \cup \mathbb{N} = \{\bullet, 0, 1, 2, \ldots\}$$

だと思っても差し支えはない．集合 $A \subseteq \mathbb{N}$ について，$\mathbf{1} + A$ も同様に，$\mathbf{1}$ と A の和集合を表すものとする．

定義 3.7. 集合 $A \subseteq \mathbb{N}$ が**計算可枚挙** (*computably enumerable*) とは，計算可能な全射 $p : \mathbb{N} \to \mathbf{1} + A$ が存在することを意味する．

豆知識． 計算可枚挙集合は，長らく**再帰的可算** (*recursively enumerable*) 集合と呼ばれてきたものである．頭文字を取って，計算量クラス RE と書かれることもあった．しかし，計算論の基礎概念について，英語圏での大規模な名称変更が前世紀末から始まり，少なくとも計算可能性理論周辺分野では，名称変更がかなり浸透している．これらの概念の和訳問題について語ることは極めて多いが，長くなるので省略する．

ところで，第 2 節では様々な決定問題を見てきた．停止問題 Halt をはじめとして，定理 2.5 の文字列書換系の到達可能性問題 $\mathsf{Reach}_{\to,v}$，定理 2.6 のモノイドの語の問題 $\mathsf{WP}_{\langle A|R\rangle}$，定理 2.8 のポストの対応問題 PCP_k，定理 2.10 の行列のモータリティ問題 MM_m はいずれも何らかの「存在」を問う．たとえば，到達可能性問題は「与えられた u に対して，u から v に辿り着くルート $u \to u_1 \to \cdots \to v$ は**存在**するか」という問題と言い換えられるし，行列のモータリティ問題は「与えられた 3×3 行列 $A_1, A_2, \ldots, A_m$ を用いて，積が零行列になる組合せは**存在**するか」という問題と言い換えられる．このような「存在」を尋ねる問題は Σ_1 であると呼ばれる．

定義 3.8. 集合 $A \subseteq \mathbb{N}$ が Σ_1 とは，ある計算可能集合 $B \subseteq \mathbb{N} \times \mathbb{N}$ が存在して，次が成立することである．

$$A = \{n \in \mathbb{N} : (\exists s \in \mathbb{N})\ (n, s) \in B\}.$$

Σ_1 という性質と，半計算可能，計算可枚挙性の関連は明らかであろう．「白いカラスが存在するか」と尋ねられたとき，我々は白いカラスを見つけた段階で捜索を停止するが，白いカラスを見つけるまでは捜索は無限に終わらない．数 n を枚挙するという動作は，枚挙ストリームのどこかに n が存在していることである．念のため，厳密に証明を与えよう．

命題 3.9. 集合 $A \subseteq \mathbb{N}$ について，以下の条件は全て同値となる．

1. A は Σ_1 集合である．
2. A は半計算可能集合である．
3. A は計算可枚挙集合である．

Proof. (1)⇒(2): A が Σ_1 ならば，対応する計算可能集合 $B \subseteq \mathbb{N} \times \mathbb{N}$ を持つ．このとき，入力 n に対して，チューリング機械 M は順に

$$(n, 0) \in B? \quad (n, 1) \in B? \quad (n, 2) \in B? \quad \ldots$$

を判定していく．$(n,s)\in B$ なる $s\in\mathbb{N}$ が見つかったときに限り $M(n)$ は停止する．このとき，

$$M(n)\downarrow \iff (\exists s\in\mathbb{N})\,(n,s)\in B \iff n\in A$$

であるから，A が半計算可能であることが示された．

(2)⇒(3): 続いて，A を半計算可能であるとすると，あるチューリング機械 M が存在して，$n\in A$ と $M(n)\downarrow$ が同値になる．定義 1.3 において，$M(n)[s]$ を $M(n)$ の計算の第 s ステップでの状況を表していたことを思い出す．このとき，

$$p(n,s)=\begin{cases} n & \text{if } M(n)[s]\downarrow \\ \bullet & \text{otherwise}\end{cases}$$

と定義する．関数 $(n,s)\mapsto M(n)[s]$ は計算可能であるから，p は計算可能である．この p が A の枚挙であることは，以下により示される．

$$n\in A \iff M(n)\downarrow \iff (\exists s)\,M(n)[s]\downarrow \iff (\exists s)\,p(n,s)=n \iff (\exists a,b)\,p(a,b)=n.$$

以上より，A が計算可枚挙集合であることが示された．

(3)⇒(1): もし $A\subseteq\mathbb{N}$ が計算可枚挙集合ならば，計算可能な全射 $p:\mathbb{N}\to\mathbf{1}+A$ が存在する．このとき，$B=\{(n,s)\in\mathbb{N}\times\mathbb{N}:p(s)=n+\bullet\}$ とする．明らかに

$$n\in A \iff (\exists s\in\mathbb{N})\,(n,s)\in B$$

であるから，A は Σ_1 である． □

第 2.5 節で紹介した MRDP 定理 2.13 を用いれば，Σ_1 集合というものは，

言語 $\{+,\cdot,=,0,1\}$ 上の一階述語論理式で全称量化 $\forall$ を使わずに定義できる $\mathbb{N}$ の部分集合

として特徴づけられる．言い換えれば，「算術の言語を用いて，存在型の式で定義できる集合」が Σ_1 集合である．実際，Σ_1 集合は通常，言語 $\{+,\cdot,\leqslant,0,1\}$ 上の述語論理式で非有界全称量化 $\forall$ を使わずに定義できる $\mathbb{N}$ の部分集合として定義されることが多い．つまり，Σ_1 集合とは，アプリオリには計算可能性概念を伴わないものであるが，それが計算可能性理論の根幹をなす概念となるのである．

■枚挙還元: 関数 $p:\mathbb{N}\to\mathbf{1}+\mathbb{N}$ が枚挙する集合を $\mathrm{Enum}(p)=\{p(n):n\in\mathbb{N}\}\cap\mathbb{N}$ によって定義する．以後，$\mathcal{P}\mathbb{N}$ によって，$\mathbb{N}$ の冪集合，つまり自然数の部分集合全体の集合を表す．

定義 3.10. 関数 $F:\mathcal{P}\mathbb{N}\to\mathcal{P}\mathbb{N}$ が**連続** (*continuous*) であるとは，次のような連続関数 $f:(\mathbf{1}+\mathbb{N})^{\mathbb{N}}\to(\mathbf{1}+\mathbb{N})^{\mathbb{N}}$ が存在することである．

$p\in(\mathbf{1}+\mathbb{N})^{\mathbb{N}}$ が $P\subseteq\mathbb{N}$ の枚挙ならば，$f(p)\in(\mathbf{1}+\mathbb{N})^{\mathbb{N}}$ は $F(P)\subseteq\mathbb{N}$ の枚挙である．

もし，この f が計算可能関数であれば，F は**枚挙作用素** (*enumeration operator*) と呼ばれる．

言い換えると，以下の図式が可換になっている．

$$\begin{array}{ccc} \mathcal{P}\mathbb{N} & \xrightarrow{F} & \mathcal{P}\mathbb{N} \\ {\scriptstyle \mathrm{Enum}}\uparrow & & \uparrow{\scriptstyle \mathrm{Enum}} \\ (\mathbf{1}+\mathbb{N})^{\mathbb{N}} & \xrightarrow[f]{} & (\mathbf{1}+\mathbb{N})^{\mathbb{N}} \end{array}$$

豆知識．位相空間論を既に学んでいる人のために述べておくと，上の意味での $\mathcal{P}\mathbb{N}$ 上の連続性は，$\mathcal{P}\mathbb{N}$ 上のカントール位相での連続性でなく，$\mathcal{P}\mathbb{N}$ 上のスコット位相での連続性であることに注意する．これは，$\mathcal{P}\mathbb{N}$ を離散空間 $\mathbf{2}$ の可算積 $\mathbf{2}^{\mathbb{N}}$ ではなくシエルピンスキ空間 $\mathbb{S}$ の可算積 $\mathbb{S}^{\mathbb{N}}$ と考えることと同値である．

$\mathbb{A}^{\mathbb{N}}$ 上の連続関数と同様に，$\mathcal{P}\mathbb{N}$ 上の枚挙作用素もまた有限的に取り扱うことができる．まず，$\mathcal{P}_{\mathrm{fin}}(\mathbb{N})$ を自然数の有限部分集合全体を表すものとすると，$\mathbb{N}$ と $\mathcal{P}_{\mathrm{fin}}(\mathbb{N})$ を一対一に対応付けられる．たとえば，有限集合 $A \subset \mathbb{N}$ と自然数 $\sum_{n\in A} 2^n$ を同一視すればよい．この自然数を有限集合 A の**標準コード** (*canonical code*) と呼ぶ．標準コード $e \in \mathbb{N}$ の有限集合は D_e と書かれる．

連続関数 $F : \mathcal{P}\mathbb{N} \to \mathcal{P}\mathbb{N}$ の**グラフコード**とは，次によって与えられる．

$$\Gamma(F) = \{\langle n, e\rangle : n \in F(\mathsf{D}_e)\}$$

逆に集合 $\Psi \subseteq \mathbb{N}$ が与えられれば，次のように連続関数 $F_\Psi : \mathcal{P}\mathbb{N} \to \mathcal{P}\mathbb{N}$ を定義できる．

$$n \in F_\Psi(A) \iff (\exists e)\,[\langle n, e\rangle \in \Psi \text{ and } \mathsf{D}_e \subseteq A].$$

命題 3.11. 枚挙作用素 F のグラフコード $\Gamma(F)$ は半計算可能である．逆に，$\Psi \subseteq \mathbb{N}$ が半計算可能ならば，F_Ψ は枚挙作用素である．

Proof. F を枚挙作用素とすると，定義 3.10 のような計算可能関数 $f : (\mathbf{1}+\mathbb{N})^{\mathbb{N}} \to (\mathbf{1}+\mathbb{N})^{\mathbb{N}}$ が存在する．このとき，定義 3.2 とその直後の注意より，ある全域計算可能単調関数 φ について $f = \widehat{\varphi}$ となる．このとき，与えられた $e \in \mathbb{N}$ に対して，有限集合 D_e の枚挙 p_e を作る機械は容易に構成できる．たとえば，D_e を小さい順に並べ上げ，全て並べ終えたら後は $\bullet$ を出力し続ければよい．つまり，ある計算可能関数 $p : \mathbb{N}\times\mathbb{N} \to \mathbf{1}+\mathbb{N}$ で，$p_e = p(e,\cdot)$ は D_e の枚挙であるようなものが存在する．f の定義より，$f(p_e)$ は $F(\mathsf{D}_e)$ の枚挙である．よって，$n \in F(\mathsf{D}_e)$ ならば，ある s について，$f(p_e)(s) = n$ となる．以上より，

$$\langle n, e\rangle \in \Gamma(F) \iff (\exists s \in \mathbb{N})\ f(p_e)(s) = n \iff (\exists s, t \in \mathbb{N})\ \varphi(p(e,0)p(e,1)\ldots p(e,t))(s) = n$$

となるから，$\Gamma(F)$ は Σ_1 である．

逆に Ψ を半計算可能であるとする．このとき，Ψ_s を Ψ の時刻 s 近似とする．つまり，命題 3.9 の証明のように，$n \in \Psi$ と $M(n)\downarrow$ が同値であるような M を取り，$\Psi_s = \{n : M(n)[s]\downarrow\}$ と定義する．与えられた有限列 $\sigma \in (\mathbf{1}+\mathbb{N})^*$ について，D_σ を σ が枚挙する有限集合，つまり $D_\sigma = \{n \in \mathbb{N} : (\exists s \in \mathbb{N})\ \sigma(s) = n \neq \bullet\}$ とする．このとき，$|\sigma|$ によって σ の長さを表すとし，

$$n \in E_\sigma \iff (\exists e)\,[\langle n, e\rangle \in \Psi_{|\sigma|} \text{ and } \mathsf{D}_e \subseteq D_\sigma]$$

と定義すると，E_σ は有限集合であり，$E = \{(n, \sigma) : n \in E_\sigma\}$ は計算可能集合である．

次によって単調関数 φ を定義する．与えられた σ に対し，$\varphi(\sigma)$ は $E_{\sigma\restriction 0}$ の要素を小さい順に枚挙し，次に $E_{\sigma\restriction 1}$ の要素を小さい順に枚挙し，$E_{\sigma\restriction 1}$ の要素を小さい順に枚挙し，... という動作を繰り返すものとする．ここで $\sigma \restriction n$ は σ の長さ n までの制限を表す．つまり，$E_\tau = \{a_0^\tau < a_1^\tau < \cdots < a_{i(\tau)}^\tau\}$ のように E_τ を下から順に並べたとき，

$$\varphi(\sigma) = \langle a_0^{\sigma\restriction 0}, a_1^{\sigma\restriction 0}, \ldots, a_{i(\sigma\restriction 0)}^{\sigma\restriction 0}, a_0^{\sigma\restriction 1}, a_1^{\sigma\restriction 1}, \ldots, a_{i(\sigma\restriction 1)}^{\sigma\restriction 1}, \ldots, a_0^\sigma, a_1^\sigma, \ldots, a_{i(\sigma)}^\sigma \rangle$$

と定義する．φ が単調関数であることは明らかである．φ の計算可能性は，E の計算可能性から従う．また，任意の A の枚挙 p に対して，$\widehat{\varphi}(p)$ が $F_\Psi(A)$ の枚挙となっていることは容易に確認できる．したがって，F_Ψ は枚挙作用素である． □

定義 3.12. 集合 $A, B \subseteq \mathbb{N}$ について，A が B に**枚挙還元** (*enumeration reducible*) されるとは，ある枚挙作用素 $\Psi : \mathcal{P}\mathbb{N} \to \mathcal{P}\mathbb{N}$ が存在して，$\Psi(B) = A$ となることである．このとき，$A \leqslant_e B$ と書く．

枚挙還元は，自然数の部分集合の持つ正の情報の複雑さを測る概念である．つまり，$A \leqslant_e B$ とは，B の正例（B を満たす例）が全て漏れずにストリームとして次々に与えられれば，A の正例を全て並べることができる，というものである．枚挙還元とチューリング還元の関連性を明確にしておこう．集合 $A, B \subseteq \mathbb{N}$ について，$A \oplus B = \{2n : n \in A\} \cup \{2n+1 : n \in B\}$ と定義する．また，$\overline{A}$ によって A の補集合，つまり $\mathbb{N}\backslash A$ を表すものとする．

命題 3.13. 任意の集合 $A, B \subseteq \mathbb{N}$ について，次が成立する．

$$A \leqslant_T B \iff A \oplus \overline{A} \leqslant_e B \oplus \overline{B}.$$

Proof. まず，任意の $X \subseteq \mathbb{N}$ について，次の 2 つの性質が成り立つことを確認する．

1. p が $X \oplus \overline{X}$ の枚挙ならば，$X \leqslant_T p$ である．
2. $p \leqslant_T X$ となるような $X \oplus \overline{X}$ の枚挙が存在する．

1 つめについて，p を $X \oplus \overline{X}$ の枚挙とする．$m \in X$ かどうかを知るためには $2m$ と $2m+1$ のどちらが p によって並べられるかを確認すればよい．よって，$X \leqslant_T p$ である．2 つめについて，

$$p(n) = \begin{cases} 2n & \text{if } n \in X \\ 2n+1 & \text{if } n \notin X \end{cases}$$

と定義すれば，p が $X \oplus \overline{X}$ の枚挙であることは容易に分かる．また，明らかに $p \leqslant_T X$ である．それでは，命題の証明を開始しよう．

($\Rightarrow$) $A \leqslant_T B$ を仮定する．$B \oplus \overline{B}$ の任意の枚挙から相対的に A が計算できることを示す．$A \leqslant_T B$ を保証する神託チューリング機械 M とすると，この M は，$A(n)$ の値を求めるために，計算中で神託に有限個のクエリ $B(m_0), B(m_1), \ldots, B(m_k)$ を作るかもしれない．したがって，$B \oplus \overline{B}$ の枚挙 p が与えられれば，上の性質 (1) より $B \leqslant_T p$ であるから，B への任意のクエリへの解を計算することができる．よって，A を計算することができる．上の性質 (2) の方法によって，$A \oplus \overline{A}$ を枚挙できる．この手続きを枚挙作用素として表すことは容易にできる．以上より，$A \oplus \overline{A} \leqslant_e B \oplus \overline{B}$ であることが示された．

($\Leftarrow$) 逆に $A \oplus \overline{A} \leqslant_e B \oplus \overline{B}$ を仮定する．このとき，$B \oplus \overline{B}$ の任意の枚挙を神託として $A \oplus \overline{A}$ のある枚挙を出力するチューリング機械 M が存在する．性質 (2) より $p \leqslant_T B$ となる $B \oplus \overline{B}$ の枚挙 p が存在する．このとき，M にストリーム p を入力すれば，$A \oplus \overline{A}$ を枚挙するストリーム q が出力される．特に $q \leqslant_T p$ である．性質 (1) より $A \leqslant_T q$ を得る．以上より，$A \leqslant_T q \leqslant_T p \leqslant_T B$ であるから，$A \leqslant_T B$ が示された． □

系 3.14. 任意の集合 $A \subseteq \mathbb{N}$ について，以下は同値である．

1. A は計算可能である．
2. A と $\overline{A}$ は共に計算可枚挙である．

Proof. A が計算可能であるということと $A \leqslant_T \varnothing$ であることは同値である．よって，命題 3.13 より，これは $A \oplus \overline{A} \leqslant_e \varnothing \oplus \mathbb{N}$ と同値である．しかし，$\varnothing \oplus \mathbb{N}$ は明らかに計算可枚挙であるから，枚挙作用素の定義 3.10 を考えれば，これは $A \oplus \overline{A}$ が計算可枚挙であることを意味する．つまり，A と $\overline{A}$ は共に計算可枚挙である． □

この系 3.14 は，ポストの定理として知られるものの特殊な形である．命題 3.9 より，計算可枚挙性は Σ_1 であることと同値であったから，つまり，$A \subseteq \mathbb{N}$ が計算可能であるということは，A と $\overline{A}$ が共に Σ_1 であるということである．これは，集合の計算可能性の算術の言語における特徴付けを与える．

ポストの定理の一つの簡単な帰結を与えよう．

系 3.15. $A \leqslant_T B$ だが $A \not\leqslant_m B$ であるような $A, B \subseteq \mathbb{N}$ が存在する．

Proof. $\mathsf{Halt}^{\mathsf{c}}$ を停止問題の補集合，つまり，$\mathsf{Halt}^{\mathsf{c}} = \{(e, n) : [\![e]\!](n) \downarrow\}$ とする．明らかに $\mathsf{Halt}^{\mathsf{c}} \leqslant_T \mathsf{Halt}$ である．一方，もし $\mathsf{Halt}^{\mathsf{c}} \leqslant_m \mathsf{Halt}$ であったと仮定すると，命題 2.3 より，$\mathsf{Halt}^{\mathsf{c}}$ は半計算可能となる．したがって，命題 3.9 より，Halt と Halt^{c} は共に計算可枚挙となるが，系 3.14 より，これは停止問題 Halt が計算可能であることを導く．これは，定理 1.31 の停止問題の計算不可能性に矛盾するから，$\mathsf{Halt}^{\mathsf{c}} \not\leqslant_m \mathsf{Halt}$ であることが分かった． □

3.3 部分組合せ代数

20 世紀初期頃から，多くの研究者が「計算可能」という概念を数学的に定式化するために，様々な計算モデルを考案してきた．代表的なものは，ラムダ計算，再帰関数，チューリング機械，あるいは世に溢れる数多のプログラミング言語である．これらの計算モデルの計算能力は全て一致することが判明し，ならばこれが計算可能性の妥当な定義であろう，ということで決着が付いた[*3]．これが**チャーチ・チューリングの提唱** (*Church-Turing thesis*) である．

ところで，結局のところ，計算可能性の本質とは何であろうか．計算可能性の原理をより単純な数学的構造として取り扱えないだろうか．ここでは，計算の本質を代数構造として抽出することを考えよう．まず，以下のような最もプレーンな代数構造を導入する．

定義 3.16. 集合 A と部分 2 項演算 $\cdot :\subseteq A \times A \to A$ の対は，**部分マグマ** (*partial magma*) と呼ばれる．部分マグマ A の元 $x, y \in A$ について，$x \cdot y$ の値が定義されているとき，$x \cdot y \downarrow$ と書く．そうでないときは，$x \cdot y \uparrow$ と書く．

注意. マグマは亜群 (groupoid) と呼ばれることもあるが，亜群 (groupoid) の名はより重要な別概念（全ての射が可逆である圏，つまり “partial group” とも呼べる代数構造）に用いられて紛らわしいので，ここでは用いない．ちなみに partial を「部分」と訳すと，sub- と紛らわしいので，この種の partial のことも「偏」と訳す流儀があるようである．たとえば，部分マグマでなく偏マグマと訳すような感じである．確かに，部分 (partial) 組合せ代数の部分 (sub) 代数を後に頻繁に取り扱うという点を考慮に入れると，偏組合せ代数と訳したい気持ちもある．

例 3.17. 任意の半群は部分マグマである．実際，半群とは，結合律 $(a \cdot b) \cdot c = a \cdot (b \cdot c)$ を満たす（全域）マグマのことである．

しかし，これから取り扱う具体的な構造は，いずれも結合律を満たさない．実際，計算論的にある種の良い性質を持つ部分マグマは，決して結合律を満たし得ない，ということが後に分かる．それでは，結合律を満たさない部分マグマには，たとえばどのようなものがあるだろうか．

例 3.18. $(\mathbb{Z}, -)$ や $(\mathbb{R}, \div)$ は非結合的な部分マグマである．たとえば，$(4 \div 2) \div 2 = 2 \div 2 = 1$ であるが，$4 \div (2 \div 2) = 4 \div 1 = 4$ である．

とはいえ，今後扱う部分マグマは，例 3.18 のような非結合的マグマとは大きく異なる．今後の話で念頭に置いておくとよい部分マグマは，関数適用のなす部分マグマである．

[*3] 一方で，20 世紀中期以降の再帰理論（計算可能性理論）のメインストリームは，計算可能性の外側の研究であり，計算可能性の外側にも計算可能性に近しい秩序と構造が広がっていることを明確にした．このような研究は 1960 年代〜80 年代にかけて，集合論やモデル理論と融合しつつ，数学的に極めて深い理論として発展し，その中で最も先鋭的な理論は，かつて**一般再帰理論** (*generalized recursion theory*) の名で隆盛を極めた．

例 3.19. Sets を集合全体のクラスとする．集合 $f \in \mathsf{Sets}$ の定義域 $\mathrm{dom}(f)$ とは，$(x, y) \in f$ となる $y \in \mathsf{Sets}$ が存在するような $x \in \mathsf{Sets}$ 全体のことである．集合 $f \in \mathsf{Sets}$ が関数であるとは，任意の $x \in \mathrm{dom}(f)$ について $(x, y) \in f$ となる $y \in \mathsf{Sets}$ が唯一であることであり，このとき $f(x) = y$ と書く．このとき，$f, x \in \mathsf{Sets}$ について，

$$f \cdot x \downarrow \iff f \text{ は関数である } \& \; x \in \mathrm{dom}(f)$$

とし，また，このときの値は $f \cdot x = f(x)$ により定義する．

部分マグマは，このような関数適用のなす代数系 $(\mathsf{Sets}, \cdot)$ を念頭に置いておくとイメージしやすい．ただし，厳密には Sets 自体は集合ではないので，気になる人は，$\mathsf{Sets} = V$ の代わりに，適当な基数 κ について V_κ を考えるとよい．

さて，計算概念の代数的抽象化の議論に戻ろう．まず，「計算モデル」というからには，基本的な関数は実装できてほしい．たとえば，各 $a \in A$ に対して，定数関数 $\mathtt{const}_a : B \to A$ を実装できるべきである．さらに言えば，$a \mapsto \mathtt{const}_a$ を実装できてほしい．つまり，$\mathtt{k}(a) = \mathtt{const}_a$ となる関数 $\mathtt{k} : A \to A^B$ を実装できる．これは以下のように表される．

$$\mathtt{k}(a)(b) = a.$$

続いて，関数適用も実装できるのがよいだろう．関数のリスト $f = (f_a : B \to C)_{a \in A}$ が与えられたとき，入力リスト $x = (x_a)_{a \in A}$, $x_a \in B$, に出力リスト $(f_a(x_a))_{a \in A}$ を対応させる関数を実装したい．これは，次の関数 $\mathtt{s} : [B \to C]^A \to [B^A \to C^A]$ が実装できるということである．

$$\mathtt{s}(f)(x)(a) = (f(a))(x(a)).$$

とりあえず，我々の「計算モデル」に求めるものは以上であるとする．

定義 3.20. 部分組合せ代数 (*partial combinatory algebra*) とは，部分マグマ $(A, \cdot)$ に**コンビネータ** (*combinator*) と呼ばれる以下の特殊な元 $\mathtt{k}, \mathtt{s} \in A$ が備わったものである．

$$\begin{aligned} (\forall a, b \in A) \quad & \mathtt{k} \cdot a \downarrow \text{ and } (\mathtt{k} \cdot a) \cdot b = a \\ (\forall f, x, a \in A) \quad & (\mathtt{s} \cdot f) \cdot x \downarrow \text{ and } ((\mathtt{s} \cdot f) \cdot x) \cdot a \simeq (f \cdot a) \cdot (x \cdot a). \end{aligned}$$

ここで，$\simeq$ は強い意味での同値性，すなわち $A \cup \{\uparrow\}$-値として一致するということである．これは，上の説明で言うところの f_a が部分関数であり，$f_a(x_a)$ が定義されない場合を想定している．さらに言えば，そもそも $a \mapsto f_a$ や $a \mapsto x_a$ が部分関数である状況も考慮に入れている．

部分組合せ代数の代数的性質を見ると，単なる部分マグマよりは幾分か秩序を持つ．たとえば，部分組合せ代数は，左単位的部分マグマである．

命題 3.21. 部分組合せ代数 A は必ず左単位元 $\mathtt{i} \in A$ を持つ．つまり，ある $\mathtt{i} \in A$ が存在して，任意の $a \in A$ に対して，$\mathtt{i} \cdot a \downarrow = a$ となる．

Proof. $\mathtt{i} = (\mathtt{s} \cdot \mathtt{k}) \cdot \mathtt{k}$ と定義する．このとき，

$$\mathtt{i} \cdot a = ((\mathtt{s} \cdot \mathtt{k}) \cdot \mathtt{k}) \cdot a \simeq (\mathtt{k} \cdot a) \cdot (\mathtt{k} \cdot a) = a$$

となるから，$\mathtt{i}$ は左単位元である． □

部分組合せ代数の重要な具体例は，チューリング機械の動作から得られる．第 1.4 節で，コード e のチューリング機械によって定義される部分関数を $[\![e]\!]$ と書いていたことを思い出そう．

命題 3.22 (クリーネの第一代数)**.** $\mathbb{N}$ 上の 2 項演算を $e \cdot n = [\![e]\!](n)$ によって定義する．このとき，$\mathbb{K}_1 = (\mathbb{N}, \cdot)$ は部分組合せ代数をなす．

Proof. $\mathbb{K}_1$ が部分組合せ代数であることを確かめるために，まず $\mathtt{k}$-コンビネータを構成しよう．まず，射影 $(u, v) \mapsto u$ は計算可能なので，$[\![i]\!](u, v) = u$ となるコード i を固定する．このとき，s をパラメータ定理 1.29 の条件を満たすものとすると，$[\![s(i, a)]\!](v) \simeq [\![i]\!](a, v) = a$ である．$a \mapsto s(i, a)$ は計算可能であるから，$[\![\mathtt{k}]\!](a) = s(i, a)$ となる $\mathtt{k} \in \mathbb{N}$ が存在する．よって，

$$[\![[\![\mathbf{k}]\!](a)]\!](b) = [\![s(i, a)]\!](b) \simeq [\![i]\!](a, b) = a.$$

つづいて，$\mathtt{s}$-コンビネータを構成しよう．まず，$(f, x, a) \mapsto [\![[\![f]\!](a)]\!]([\![x]\!](a))$ は計算可能なので，このコードを e とする．s をパラメータ定理 1.29 の条件を満たすものとすると，$[\![s(e, f, x)]\!](a) \simeq [\![e]\!](f, x, a)$ である．$(f, x) \mapsto s(e, f, x)$ は計算可能であるから，再びパラメータ定理 1.29 より，$[\![[\![\mathtt{s}]\!](f)]\!](x) \simeq s(e, f, x)$ なる $\mathtt{s} \in \mathbb{N}$ が存在する．よって，

$$[\![[\![[\![\mathtt{s}]\!](f)]\!](x)]\!](a) \simeq [\![s(e, f, x)]\!](a) \simeq [\![e]\!](f, x, a) \simeq [\![[\![f]\!](a)]\!]([\![x]\!](a)).$$

以上より，$\mathbb{K}_1$ が部分組合せ代数であることが示された． □

命題 3.22 の部分組合せ代数 $\mathbb{K}_1 = (\mathbb{N}, \cdot)$ は**クリーネの第一代数** (*Kleene's first algebra*) と呼ばれる．自然数と有限語は同一視できるので，$\mathbb{K}_1 = (\Sigma^*, \cdot)$ と思ってもよい．他にも部分組合せ代数の具体例は多数あるので，そのうちの重要な例をいくつか紹介する．

例 3.23 (クリーネの第二代数)**.** 各 $p : \mathbb{N}^* \times \mathbb{N} \to (\mathbf{1} + \mathbb{N})$ は単調関数 $\varphi_p : \mathbb{N}^* \to \mathbb{N}^*$ を以下のように生成する．ここで，$\mathbf{1} = \{\bullet\}$ である．$\varphi_p(\sigma)$ は帰納的に定義される．もし任意の $m < n$ について $\varphi_p(\sigma)(m)$ が定義されているならば，$\varphi_p(\sigma)(n)$ を以下のように定義する．

$$\varphi_p(\sigma)(n) = \begin{cases} p(\tau, n), & \text{ここで } \tau \text{ は } p(\tau, n) \neq \bullet \text{ なる最初の } \tau \preceq \sigma \text{ である．} \\ \text{未定義}, & \text{そのような } \tau \text{ が存在しない．} \end{cases}$$

$\mathbb{N} \simeq \mathbb{N}^* \times \mathbb{N} \simeq \mathbf{1} + \mathbb{N}$ という同一視をすれば，$p \in \mathbb{N}^{\mathbb{N}}$ と思える．つまり，任意の $p \in \mathbb{N}^{\mathbb{N}}$ は上のように部分単調関数 $\varphi_p :\subseteq \mathbb{N}^* \to \mathbb{N}^*$ を定義し，よって，定義 3.1 のように部分連続関数 $\widehat{\varphi_p} :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ を定義する．$\mathbb{N}^{\mathbb{N}}$ 上の 2 項演算を $p \cdot x = \widehat{\varphi_p}(x)$ によって定義する．このとき，$\mathbb{K}_2 = (\mathbb{N}^{\mathbb{N}}, \cdot)$ は**クリーネの第二代数** (*Kleene's second algebra*) と呼ばれ，部分組合せ代数をなす．

例 3.24 (スコットのグラフモデル)**.** 第 3.2 節で見たように，$A \subseteq \mathbb{N}$ は連続関数 $F_A : \mathcal{P}\mathbb{N} \to \mathcal{P}\mathbb{N}$ を定義する．$\mathcal{P}\mathbb{N}$ 上の 2 項演算を $A \cdot B = F_A(B)$ によって定義する．このとき，$\mathbb{P} = (\mathcal{P}\mathbb{N}, \cdot)$ は**スコットのグラフモデル** (*Scott's graph model*) と呼ばれ，部分組合せ代数をなす．グラフモデル $\mathbb{P}$ における 2 項演算は全域であるため，**全域組合せ代数** (*total combinatory algebra*) と呼ばれることもある．

ところで，クリーネの第一代数 $\mathbb{K}_1$ の演算は常に計算可能関数によって実装される．しかし，クリーネの第二代数 $\mathbb{K}_2$ とスコットのグラフモデル $\mathbb{P}$ では，各演算の実装が連続関数による．計算可能性理論の観点からは，演算適用はそれぞれ計算可能連続関数と枚挙作用素であってほしい．このため，相対部分組合せ代数の概念を導入する．

定義 3.25. 相対部分組合せ代数 (*relative partial combinatory algebra*) とは，2 項演算とコンビネータ $\mathtt{k}, \mathtt{s}$ を共有する部分組合せ代数の対 $\mathbf{A} = (\underset{\sim}{\mathbf{A}}; \mathrm{A})$ で，$\mathtt{k}, \mathtt{s} \in \mathrm{A} \subseteq \underset{\sim}{\mathbf{A}}$ かつ，以下を満たすものである．

$$a, b \in \mathrm{A} \text{ and } a \cdot b \downarrow \in \underset{\sim}{\mathbf{A}} \implies a \cdot b \in \mathrm{A}.$$

このとき，$\underset{\sim}{\mathbf{A}}$ 上の部分関数 $f :\subseteq \underset{\sim}{\mathbf{A}} \to \underset{\sim}{\mathbf{A}}$ が **A-実現可能** (**A**-*realizable*) であるとは，以下を満たすことである．

$$(\exists e \in \underset{\sim}{\mathbf{A}})(\forall a \in \mathrm{dom}(f))\ [e \cdot a \downarrow, \text{ and } e \cdot a = f(a)].$$

$e \in \mathrm{A}$ として取れる場合，f は **A-計算可能** (**A**-*computable*) であるという．

相対部分組合せ代数の概念を説明すると，我々は計算可能なものと計算不可能なものが混在する世界に住んでいるものと考えよう．$\underset{\sim}{\mathbf{A}}$ がそのような世界を表し，A はそのうちの計算可能なものだけを取り出したものである．これはたとえば，記述集合論における**太字** (*boldface*) と**細字** (*lightface*) のポイントクラスと類似の発想である．

例 3.26. $\mathbb{K}_1^r = (\mathbb{K}_1, \mathbb{K}_1)$ は相対部分組合せ代数である．このとき，$\mathbb{K}_1^r$-実現可能関数および $\mathbb{K}_1^r$-計算可能関数とは，$\mathbb{N}$ 上の部分計算可能関数である．

例 3.27. クリーネの第二代数 $\mathbb{K}_2$ について，コンビネータ $\mathtt{k}$ と $\mathtt{s}$ は $\mathbb{N}^{\mathbb{N}}$ の要素であるが，これを $\mathbb{N}$ 上の関数と思うと，$\mathtt{k}$ と $\mathtt{s}$ を計算可能関数として選ぶことができる．$\Delta_1^0 \subseteq \mathbb{N}^{\mathbb{N}}$ で $\mathbb{N}$ 上の計算可能関数全体を表すとする．このとき，$\mathbb{K}_2^\circ = (\Delta_1^0, \cdot, \mathtt{k}, \mathtt{s})$ を考えると，$\mathbb{K}_2^r = (\mathbb{K}_2, \mathbb{K}_2^\circ)$ は相対部分組合せ代数をなす．このとき，$\mathbb{K}_2^r$-実現可能関数および $\mathbb{K}_2^r$-計算可能関数とは，それぞれ $\mathbb{N}^{\mathbb{N}}$ 上の部

分連続関数および部分計算可能関数である．

例 3.28. スコットのグラフモデル $\mathbb{P}$ について，コンビネータ $\mathtt{k}$ と $\mathtt{s}$ を計算可枚挙集合として選ぶことができる．$\Sigma^0_1 \subseteq \mathcal{P}\mathbb{N}$ で $\mathbb{N}$ 上の計算可枚挙集合全体を表すとする．このとき，$\mathbb{P}^\circ = (\Sigma^0_1, \cdot, \mathtt{k}, \mathtt{s})$ を考えると，$\mathbb{P}^r = (\mathbb{P}, \mathbb{P}^\circ)$ は相対部分組合せ代数をなす．このとき，$\mathbb{P}^r$-実現可能関数および $\mathbb{P}^r$-計算可能関数とは，それぞれ $\mathcal{P}\mathbb{N}$ 上の連続関数および枚挙作用素である．

相対部分組合せ代数は対であるが，単に $\mathbf{A}$ と表すことも多い．その場合，常に定義 3.25 のような $\mathrm{A} \subseteq \underset{\sim}{\mathbf{A}}$ を伴うものと考える．$\underset{\sim}{\mathbf{A}} = \mathrm{A}$ であるような相対部分組合せ代数は**フル** (*full*) であると呼ばれる．たとえば，$\mathbb{K}^r_1$ はフルだが，$\mathbb{K}^r_2$ と $\mathbb{P}^r$ はフルではない．

定義 3.29. 相対部分組合せ代数 $\mathbf{A}$ が与えられているとする．$a, b \in \mathbf{A}$ について，ある部分 $\mathbf{A}$-計算可能関数 $f :\subseteq \mathbf{A} \to \mathbf{A}$ が存在して，$f(b) = a$ となるとき，a は b から**相対的 $\mathbf{A}$-計算可能** (*relatively* $\mathbf{A}$*-computable*) であると言い，$a \leqslant_{\mathbf{A}} b$ と書く．つまり，

$$a \leqslant_{\mathbf{A}} b \iff (\exists e \in \mathrm{A})\, e \cdot b \downarrow = a$$

によって定義する．

例 3.30. クリーネ第二代数 $\mathbb{K}^r_2$ における相対的計算可能性 $\leqslant_{\mathbb{K}^r_2}$ はチューリング還元 $\leqslant_T$ と同値である．スコットのグラフモデル $\mathbb{P}^r$ における相対的計算可能性 $\leqslant_{\mathbb{P}^r}$ は枚挙還元 $\leqslant_e$ と同値である．

3.4 ラムダ計算，不動点，再帰定理

これから，如何なる部分組合せ代数の中でも “計算” を展開できることを示す．つまり，あらゆる計算を，積の適用という代数的演算の組合せで表せる，ということを見る．以後，部分組合せ代数が与えられたときに，積 $\cdot$ の記号は省略し，また積は左結合的であると仮定する．つまり，$(a \cdot b) \cdot c$ は abc のように積と括弧は省略する．たとえば，

$$\mathtt{k}ab = a, \quad \mathtt{s}abc \simeq ac(bc), \quad \mathtt{i} = \mathtt{skk}$$

のように書かれる．

定義 3.31. $\mathbf{A}$ を部分組合せ代数とする．$\mathbf{A}$ 上の**項**とは以下のように帰納的に定められる．

1. 変数 $x, y, z, \ldots$ は項である．
2. 各元 $a \in \mathbf{A}$ は項である．
3. P と Q が項ならば PQ も項である．

A 上の項 P と変数 x が与えられたとき，項 $\Lambda x.P$ を以下によって帰納的に定義する．

$$\begin{aligned} \Lambda x.x &\equiv \mathtt{i} \\ \Lambda x.y &\equiv \mathtt{k}y \qquad (y \neq x \text{ が変数であるか } y \in \mathbf{A} \text{ のとき}) \\ \Lambda x.(PQ) &\equiv \mathtt{s}(\Lambda x.P)(\Lambda x.Q) \end{aligned}$$

ここで $\mathtt{i}$ は命題 3.21 のような **A** の左単位元とする．

以後，$\Lambda x.(\Lambda y.(\Lambda z.P))$ などは $\Lambda xyz.P$ のように省略する．たとえば，

$$\Lambda xy.x = \Lambda x.(\Lambda y.x) = \Lambda x.(\mathtt{k}x) = \mathtt{s}(\Lambda x.\mathtt{k})(\Lambda x.x) = \mathtt{s}(\mathtt{kk})\mathtt{i}$$

となる．項 $\Lambda x.P$ に含まれる変数は，項 P に含まれる変数から x を除いたものである．よって，P が x のみを変数として含むならば，$\Lambda x.P$ は変数を含まない．定義 3.20 より，$\mathtt{k}a\downarrow$ および $\mathtt{s}ab\downarrow$ が成立しているから，このとき $\Lambda x.P$ は **A** の元を定義することが示される．

項 P が与えられたとき，$P[Q/x]$ によって，P の変数 x に項 Q を代入した結果を表す．上の議論をより一般化すると，$y_0, \ldots, y_n$ を $\Lambda x.P$ に含まれる自由変数のリストとすれば，任意の $a_0, \ldots, a_n \in \mathbf{A}$ について，$(\Lambda x.P)[a_0, \ldots, a_n/y_0, \ldots, y_n]$ は **A** の元を表す．

以下，重要な性質として，$\Lambda x.P$ は本物のラムダ計算のような働きをするということを示す．つまり，部分組合せ代数が与えられれば，内部でラムダ計算っぽいものを展開できてしまうということである．

補題 3.32. P と Q を **A** 上の項とする．このとき，$(\Lambda x.P)Q \simeq P[Q/x]$ が成り立つ．

Proof. P の構造に関する帰納法による．$P = x$ のとき，

$$(\Lambda x.P)Q = \mathtt{i}Q = Q = P[Q/x].$$

続いて，$P = y$ が $y \neq x$ なる変数であるか，$y \in \mathbf{A}$ であるとき，

$$(\Lambda x.P)Q = \mathtt{k}yQ = y = P[Q/x].$$

最後に，$P = RS$ である場合，帰納的に $(\Lambda x.R)Q \simeq R[Q/x]$ かつ $(\Lambda x.S)Q \simeq S[Q/x]$ が成り立っていると仮定する．このとき，

$$(\Lambda x.P)Q = \mathtt{s}(\Lambda x.R)(\Lambda x.S)Q \simeq (\Lambda x.R)Q((\Lambda x.S)Q) \simeq R[Q/x]S[Q/x] = (RS)[Q/x] = P[Q/x].$$

以上より，帰納法によって，求める性質が得られる． □

命題 3.33. 任意の部分組合せ代数 $\mathbf{A}$ について，次を満たす $\mathtt{pair}, \pi_0, \pi_1 \in \mathbf{A}$ が存在する．

$$\mathtt{pair}\, ab \downarrow, \quad \pi_0(\mathtt{pair}\, ab) = a, \quad \pi_1(\mathtt{pair}\, ab) = b$$

Proof. $\mathtt{pair} = \Lambda xyz.zxy$, $\pi_0 = \Lambda z.z(\Lambda xy.x)$, $\pi_1 = \Lambda z.z(\Lambda xy.y)$ によって定義する．このとき，補題 3.32 を利用すると，Λ は代入と解釈できるから，$\mathtt{pair}\, ab = \Lambda z.zab$ となる．したがって，補題 3.32 より，

$$\begin{aligned}\pi_0(\mathtt{pair}\, ab) &= (\Lambda z.z(\Lambda xy.x))(\Lambda z.zab) = (\Lambda z.zab)(\Lambda xy.x) = (\Lambda xy.x)ab = a,\\ \pi_1(\mathtt{pair}\, ab) &= (\Lambda z.z(\Lambda xy.y))(\Lambda z.zab) = (\Lambda z.zab)(\Lambda xy.y) = (\Lambda xy.y)ab = b.\end{aligned}$$

よって，主張は示された． □

以後，$\mathtt{pair}\, ab$ のことを $\langle a, b\rangle$ と書くことにする．このとき，$\mathbf{A}$ の任意の有限列 $(a_i)_{i\leqslant n}$ は，次のように 1 つの要素としてコードできる．

$$\langle a_0, a_1, a_2, \ldots, a_n\rangle := \langle\langle\langle\langle a_0, a_1\rangle, a_2\rangle, \ldots\rangle, a_n\rangle.$$

それでは，部分組合せ代数が与えられれば，自然数上の計算論が展開できることを見ていこう．このために，部分組合せ代数 $\mathbf{A}$ の中で自然数をコードする必要がある．これはたとえば次のようにコードできる．

定義 3.34. $\mathbf{A}$ を部分組合せ代数とする．$\mathtt{true}, \mathtt{false} \in \mathbf{A}$ および各自然数 $n \in \mathbb{N}$ について $\underline{n} \in \mathbf{A}$ を以下のように定義する．

$$\begin{aligned}\mathtt{true} &= \Lambda xy.x & \mathtt{false} &= \Lambda xy.y\\ \underline{0} &= \langle \mathtt{true}, \mathtt{i}\rangle & \underline{n+1} &= \langle \mathtt{false}, \underline{n}\rangle\end{aligned}$$

この自然数のコード方法を理解しかねるという人もいるかもしれないが，取り扱いの容易さから，このコーディングを利用する．しかし，「自然数を実装できる」という点のみが重要なのであって，自然数の具体的な実装方法は何でもよいし，その意味を問うことはあまり生産的ではない．文字列によって自然数をコードする方法がいくらでもあるように，部分組合せ代数 $\mathbf{A}$ の中で自然数をコードする方法は唯一ではない．

命題 3.35. 次のような元 $\mathtt{succ}, \mathtt{pred}, \mathtt{iszero} \in \mathbf{A}$ が存在する．

$$\mathtt{succ}\, \underline{n} = \underline{n+1} \qquad \mathtt{pred}\, \underline{n} = \underline{n \dot{-} 1}$$

$$\mathtt{iszero}\, \underline{n} = \begin{cases}\mathtt{true} & \text{if } n = 0\\ \mathtt{false} & \text{if } n \neq 0\end{cases}$$

ここで $\dot{-}$ は $x \dot{-} y = \max\{0, x - y\}$ によって定義される部分的減法である．

Proof. まず，$\mathtt{succ} = \Lambda x.\langle \mathtt{false}, x\rangle$ によって定義する．このとき，補題 3.32 より，$\mathtt{succ}\,\underline{n} = \langle \mathtt{false}, \underline{n}\rangle = \underline{n+1}$ である．$\mathtt{iszero} = \pi_0$ が条件を満たすことは明らかである．最後に，$\mathtt{pred}$ を定義するために，補題 3.32 から以下の式を得られることに注意する．

$$\langle a,b\rangle\mathtt{true} = (\Lambda xyz.zxy)ab(\Lambda xy.x) = (\Lambda z.zab)(\Lambda xy.x) = (\Lambda xy.x)ab = a$$
$$\langle a,b\rangle\mathtt{false} = (\Lambda xyz.zxy)ab(\Lambda xy.y) = (\Lambda z.zab)(\Lambda xy.y) = (\Lambda xy.y)ab = b.$$

$\mathtt{pred} = \Lambda x.\langle \underline{0}, \pi_1 x\rangle(\mathtt{iszero}\,x)$ と定義する．このとき，上の式と補題 3.32 より，

$$\mathtt{pred}(\underline{0}) = \langle \underline{0}, \pi_1\,\underline{0}\rangle(\mathtt{iszero}\,\underline{0}) = \langle \underline{0}, \mathtt{i}\rangle\mathtt{true} = \underline{0}$$
$$\mathtt{pred}(\underline{n+1}) = \langle \underline{0}, \pi_1\,\underline{n+1}\rangle(\mathtt{iszero}\,\underline{n+1}) = \langle \underline{0}, \underline{n}\rangle\mathtt{false} = \underline{n}.$$

よって，主張は示された． □

次に，部分組合せ代数におけるある種の不動点定理を示そう．関数 $f : X \to X$ の**不動点** (*fixed point*) とは，$f(x) = x$ なる $x \in X$ のことである．しかし，ここでは X は関数空間 $[A \to B]$ であると考えると都合がよい．つまり，関数 $f : [A \to B] \to [A \to B]$ の不動点とは $g = f(g)$ を満たす関数 $g : A \to B$ のことである．f の不動点のことを $\mathtt{fix}\,f$ と書くとすると，$\mathtt{fix}\,f = f(\mathtt{fix}\,f)$ を満たす．つまり

$$(\forall a \in A)\ f(\mathtt{fix}\,f)(a) = (\mathtt{fix}\,f)(a)$$

を満たすということである．このような不動点 $\mathtt{fix}\,f$ のようなものが常に存在する，というのが次の定理である．

定理 3.36. 次のような元 $\mathtt{fix} \in \mathbf{A}$ が存在する．任意の $f, a \in \mathbf{A}$ に対して，

$$\mathtt{fix}\,f \downarrow \qquad \mathtt{fix}\,fa = f(\mathtt{fix}\,f)a.$$

Proof. $\mathtt{r} = \Lambda xyz.x(yy)z$ とし，$\mathtt{fix} = \Lambda g.\mathtt{r}g(\mathtt{r}g)$ と定義する．このとき，

$$\mathtt{fix}\,f = \mathtt{r}f(\mathtt{r}f) = (\Lambda xyz.x(yy)z)f(\mathtt{r}f) = (\Lambda yz.f(yy)z)(\mathtt{r}f) = \Lambda z.f(\mathtt{r}f(\mathtt{r}f))z$$

である．いま，$\mathtt{r}$ は自由変数を持たず，よって，$f(\mathtt{r}f(\mathtt{r}f))z$ は z のみを自由変数に含む．一方，定義 3.31 の直後に述べたように，もし P が z のみを変数に含むならば，$\Lambda z.P \downarrow$ である．よって，$\mathtt{fix}\,f \downarrow$ を得る．また，

$$\mathtt{fix}\,fa = (\mathtt{r}f(\mathtt{r}f))a = (\Lambda z.f(\mathtt{r}f(\mathtt{r}f))z)a = f(\mathtt{r}f(\mathtt{r}f))a = f(\mathtt{fix}\,f)a$$

となるから定理は示された． □

不動点定理 3.36 の重要な帰結の 1 つが，**クリーネの再帰定理** (*Kleene's recursion theorem*) である．まず，定理 3.36 から次の系が得られる．

系 3.37. $\mathbf{A}$ を部分組合せ代数とし，Q を e のみを自由変数とする $\mathbf{A}$ 上の項とする．このとき，次を満たす項 $r \in \mathbf{A}$ が存在する．

$$(\forall a \in \mathbf{A})\ ra \simeq Q[r/e]a.$$

Proof. $f = \lambda e.Q$ とする．ここで，Q に含まれる自由変数は e のみであるから，定義 3.31 の直後の議論より，$f \downarrow \in \mathbf{A}$ と考えてよい．よって，不動点定理 3.36 の証明の $\mathtt{fix}$ について，$\mathtt{fix}\, f \downarrow = r$ となる $r \in \mathbf{A}$ を得る．いま，任意の $a \in \mathbf{A}$ について，

$$ra = fra = (\lambda e.Q)ra \simeq Q[r/e]a$$

であるから，主張は示された． □

ここで，$r = Q[r/e]$ となるとは限らないことに注意する．これは，クリーネの第一代数 $\mathbb{K}_1$ で考えたときに明確となる．$\mathbb{K}_1$ において，$p = q$ であるということは，この 2 つの p と q が一字一句違わない文字列あるいはプログラムであることを意味する．一方，全ての $a \in \mathbb{K}_1$ について $pa \simeq qa$ ということは，この 2 つの p と q がプログラムとして同一の働きをすることを意味している．後者の概念の方が本質的であり，前者を要求することはあまり重要でないということは予想が付くだろう．系 3.37 を $\mathbb{K}_1$ で解釈した結果は，クリーネの再帰定理として知られる．

定理 3.38 (クリーネの再帰定理)**.** 任意の計算可能関数 $q : \mathbb{N} \to \mathbb{N}$ について，次を満たす $r \in \mathbb{N}$ が存在する．

$$[\![r]\!] \simeq [\![q(r)]\!]$$

Proof. 部分組合せ代数として，命題 3.22 のクリーネの第一代数 $\mathbb{K}_1$ を取る．このとき，q は計算可能であるから，ある $d \in \mathbb{N}$ について，$q = [\![d]\!]$ となる．$Q = de$ とすると，系 3.37 より，任意の $a \in \mathbb{K}_1$ について，$ra \simeq Q[r/e] \simeq dra$ なる $r \in \mathbb{K}_1$ を得る．$\mathbb{K}_1$ における積演算の定義より，

$$(\forall a \in \mathbb{N})\ [\![r]\!](a) \simeq [\![[\![d]\!](r)]\!](a) \simeq [\![q(r)]\!](a)$$

となるから，定理は示された． □

クリーネの再帰定理の直感的な説明を与えよう．固定した未知変数 I を使いながら，コンピュータ・プログラム $Q(I)$ を書いている，というシチュエーションを想定しよう．我々はその段階では I が何であるかは知らないが，とにかく I は自由に使えるので，プログラム内部に「プログラム I に n を入力した計算を実行せよ」などの命令を書き込むことができる．

さて，クリーネの再帰定理 3.38 における r を取ってきて，$Q = Q(r)$ としよう．つまり，先ほど我々の書いたプログラムの中で I と書かれている部分を r で上書きしたものが新しいプログラム Q である．すると，再帰定理より，コード r のプログラムを実行したものと Q を実行したものの計算結果は等しい．したがって，Q のプログラム内部に書かれている「プログラム I に n を入力した計算を実行せよ」という命令は「プログラム Q に n を入力した計算を実行せよ」という命令に等しい．

これが意味していることは何だろうか．我々はプログラム Q を書いている途中段階では，最終的な Q がどうなるかは知らないし，無限の自由度がある．それにも関わらず，我々が途中で書いた「プログラム I に n を入力した計算を実行せよ」と言う命令の意味は，常に「最終的な Q に n を入力した計算を実行せよ」を表す．つまり，我々はあたかも「最終的に書き上げる予定のプログラムが何であるか既に知っているかの如く」プログラムを記述することができるのである．

そういうわけで，以後，プログラム I すなわち『私』すなわち自己に言及したプログラムは自由に記述してよい．特に自身のソースコードを出力するプログラムは，コンピュータ・プログラミングの文脈では**クワイン** (*Quine*) としてよく知られており，様々なプログラミング言語での実装例を見つけることができるだろう．

クリーネの再帰定理は数学的には一見単純であるが，それ故に強力である．基礎的なレベルから研究の最先端に至るまで，極めて広範な応用を持つ．計算可能性理論の入門的内容における最も重要な定理と言っても過言ではないだろう．

ここでは，クリーネの再帰定理の簡単な応用として，原始再帰法の実装を行う．原始再帰法とは，関数 f, g から次のような関数 h を作る操作である．

$$h(0, x) = g(x)$$
$$h(n+1, x) = f(n, x, h(n, x))$$

この関数 h を $\mathtt{rec}\, gf$ として表そう．この原始再帰作用素 $\mathtt{rec}$ は，不動点として実装できる．

命題 3.39. 次のような元 $\mathtt{rec} \in \mathbf{A}$ が存在する．

$$\mathtt{rec}\, gf\underline{0} = g$$
$$\mathtt{rec}\, gf\underline{n+1} = f\underline{n}(\mathtt{rec}\, gf\underline{n})$$

Proof. まず，命題 3.35 の証明で見たように，

$$\langle a, b\rangle \mathtt{iszero}\, n = \begin{cases} a & \text{if } n = \underline{0} \\ b & \text{if } n \neq \underline{0} \end{cases}$$

が成り立つことに注意する．このため，$\langle a, b\rangle \mathtt{iszero}\, n$ を $\mathtt{if}\ n\ \mathtt{iszero}\ \mathtt{then}\ a\ \mathtt{else}\ b$ と表す．

証明のアイデアとしては，$n = 0$ ならば $hn = g$ であり，さもなくば $hn = f(\mathtt{pred}\, n)(h(\mathtt{pred}\, n))$ であるような h を作ればよい．ただし，h は f と g に依存するので，$h = egf$ という形である．具体的には，次の項 Q を考える．

$$\Lambda gfn.\mathtt{if}\ n\ \mathtt{iszero}\ \mathtt{then}\ g\ \mathtt{else}\ f(\mathtt{pred}\, n)(egf(\mathtt{pred}\, n)).$$

項 Q は e のみを自由変数に持つので，系 3.37 より，e を自己への言及と解釈するような $\mathtt{rec} \in \mathbf{A}$ が存在する．つまり，任意の $a \in \mathbf{A}$ について $\mathtt{rec}\, a = Q[\mathtt{rec}/e]a$ が成立する．このとき，

$$\mathtt{rec}\, gf\underline{n} = \mathtt{if}\ \underline{n}\ \mathtt{iszero}\ \mathtt{then}\ g\ \mathtt{else}\ f(\mathtt{pred}\, \underline{n})(\mathtt{rec}\, gf(\mathtt{pred}\, \underline{n}))$$

であるから，明らかに

$$\mathtt{rec}\, g f \underline{0} = g, \qquad\qquad \mathtt{rec}\, g f \underline{n+1} = f\underline{n}(\mathtt{rec}\, g f \underline{n})$$

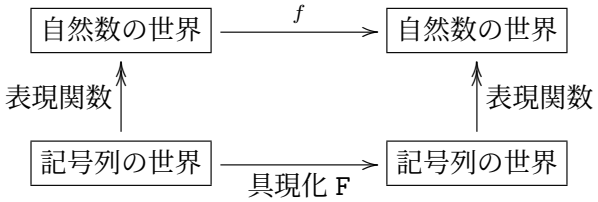
が導かれる．よって求める性質が示された． □

4 表現空間の理論

計算理論は，記号列によってコードされた数学的オブジェクトに対する計算を取り扱う分野である．この節では，文字列以外の数学的対象に関する様々な計算論を統一的に導入することを試みる．まず，自然数を記号列によって**表現**（コーディング）する，ということに立ち返って，表現（コーディング）とは何であるかの再理解を目指そう．

チューリング機械は，基本的には語上の関数 $f : (\Sigma^*)^k \to \Sigma^*$ の関数を議論するものである．しかし，2 進表現 bin によるコーディングを経由することによって，自然数上の関数 $f : \mathbb{N}^k \to \mathbb{N}$ の計算理論を展開できることは，これまでに見た通りである．これは，文字列 $\sigma \in \Sigma^*$ がどんな自然数を表しているのか，という**意味**を与えることにより，ただの文字列上の関数に自然数上の関数としての意味を与えていた，ということである．もう少し正確には，文字列 $\mathtt{bin}(n)$ が自然数 n を**表現**していたのである．この（部分）全射 $\mathtt{bin}(n) \mapsto n$ を**表現関数**と呼ぶことにしよう．

たとえば，自然数上の関数 $f : \mathbb{N}^k \to \mathbb{N}$ が計算可能であるとは，（表現関数 $\mathtt{bin}(n) \mapsto n$ を介することにより）それを記号列上の機械的操作 $\mathtt{F} :\subseteq (\Sigma^*)^k \to \Sigma^*$ として**具現化**できるということである．

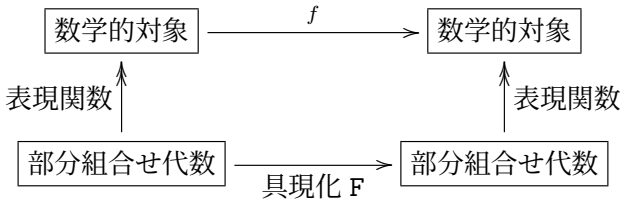


このような「**表現と具現化を経由した計算**」が計算理論の基本である．有理数や代数的数は容易に自然数でコードできるから，そのような数に関する計算論も展開できる．また，その他にも幾らかの単純な可算構造ならばコーディングできそうである．それでは，実際，どのような数学的対象ならば，記号的に表現でき，そして良い計算論を展開できるだろうか．

歴史的には，1950 年代頃に，語あるいは自然数によるコーディングという概念自体を研究対象とする**ナンバリングの理論** (*Theory of numbering*) が誕生する．また，それとは並行に，20 世紀中頃から，実数などの連続的概念を対象とする計算論である**計算可能解析学** (*computable analysis*) という分野も徐々に発展を見せた．

計算理論とはデジタル（離散的）な概念を対象とするものであり，実数や複素数などの連続的概念は計算理論の対象外である……と思われがちだが，実際はそうではない．解析学と物理学における計算可能性理論の教科書として，1989 年に Pour-El と Richards によって執筆された “Computability in Analysis and Physics” は非常に有名である

それでは，このような連続的オブジェクトを含む様々な数学的対象は如何にして計算論的に取り扱われているだろうか．その 1 つの解答は，表現と具現化に関する上の図式に少々の修正を加えるだけである．我々の記号の世界は，任意の**部分組合せ代数**であり，これによって様々な数学的対象の上の計算は，部分組合せ代数の単なる演算適用として実現される．



まず，この図式の縦方向，すなわち，部分組合せ代数による数学的対象の表現は，以下のように定式化される．

定義 4.1. $\mathbf{A}$ を (相対) 部分組合せ代数とする．このとき, 集合 X と部分全射 $\nu_X :\subseteq \mathbf{A} \to X$ の対 (X, ν_X) を **$\mathbf{A}$-表現空間** ($\mathbf{A}$-*represented space*) と呼び，ν_X を X の **$\mathbf{A}$-表現** ($\mathbf{A}$-*representation*) と呼ぶ.

これは，集合 X の各要素が，部分組合せ代数 $\mathbf{A}$ の元によって名付けられたことを意味する．つまり，$\nu_X(a) = x$ であるとき，$a \in \mathbf{A}$ は x の **ν_X-コード** (*code*) または **ν_X-名** (*name*) と呼ばれる．表現関数 ν_X が文脈から明らかな場合は，単にコードまたは名と呼ぶ．1 つの元 $x \in X$ が複数のコードを持ち得る場合もあることに注意する．$\nu_X(a)$ の代わりにしばしば $[a]_X$ と書くことがある．このとき，各 $x \in X$ について，$||x||_X$ によって x のコード全体の集合を表す．つまり，以下のように定義する．

$$||x||_X = \{a \in \mathbf{A} : x = [a]_X\}.$$

注意．実現可能性理論 (*realizability theory*) では，$(X, ||\cdot||_X)$ は**モデスト集合** (*modest set*) と呼ばれる．本稿では，しばしば表現空間 $(X, [\cdot]_X)$ とモデスト集合 $(X, ||\cdot||_X)$ を同一視し，後者のことも表現空間と呼ぶ.

例 4.2. 部分組合せ代数 $\mathbf{A}$ の中で常に自然数をコードできることは定義 3.34 において見た通りである．これを表現空間の言葉で言い直そう．定義 3.34 では，自然数 $n \in \mathbb{N}$ に対して，対応する $\underline{n} \in \mathbf{A}$ を具体的に与えた．部分関数 $\nu_{\mathbb{N}} :\subseteq A \to \mathbb{N}$ を $\nu_{\mathbb{N}}(\underline{n}) \to n$ によって定義すれば，$(\mathbb{N}, \nu_{\mathbb{N}})$ は $\mathbf{A}$-表現空間をなす．各 $n \in \mathbb{N}$ の $\nu_{\mathbb{N}}$-コードは $\underline{n} \in A$ のみであり，よって $||n||_{\mathbb{N}} = \{\underline{n}\}$ である.

豆知識．表現が多価である場合もしばしば重要となる．集合 X と部分多価全射 $\delta_X :\subseteq \mathbf{A} \rightrightarrows X$ の対 (X, δ_X) を **$\mathbf{A}$-多価表現空間** ($\mathbf{A}$-*multi-represented space*) と呼ぶ．実現可能性理論においては，これと同一な概念は **アセンブリ** (*assembly*) と呼ばれる.

つづいて，上の図式における，数学的対象上の写像の部分組合せ代数上の関数による具現化の部分である．これは，以下のように定式化される.

定義 4.3. $\mathcal{X} = (X, \nu_X)$ と $\mathcal{Y} = (Y, \nu_Y)$ を $\mathbf{A}$-表現空間とする．このとき，$\mathtt{F} :\subseteq \mathbf{A} \to \mathbf{A}$ が $f : \mathcal{X} \to \mathcal{Y}$ の**具現化**であるとは，任意の $x \in X$ に対して，x のどんな ν_X-コード $\mathtt{x} \in \mathbf{A}$ が与えられても，$\mathtt{F}(\mathtt{x})$ が $f(x)$ の ν_Y-コードを与えていることを意味する．

つまり，$\mathtt{F}$ が f の具現化であるとは，以下の図式を可換にすることである．

$$\begin{array}{ccc} X & \xrightarrow{f} & Y \\ {\scriptstyle \nu_X}\uparrow & & \uparrow{\scriptstyle \nu_Y} \\ \mathbf{A} & \xrightarrow[\mathtt{F}]{} & \mathbf{A} \end{array}$$

ここで，具現化 $\mathtt{F}$ として，部分組合せ代数上の関数であれば何でもよい．しかし，計算理論としては，$\mathtt{F}$ としては，部分組合せ代数上の演算として実現できるものであってほしい．ここで，定義 3.25 における実現可能関数と計算可能関数のことを思い出そう．これによって，空間上の関数で，記号操作によって実現できるものと，計算できるもの，そうでないものが明確に分けられる．

定義 4.4. $\mathcal{X}$ と $\mathcal{Y}$ を $\mathbf{A}$-表現空間とする．関数 $f : \mathcal{X} \to \mathcal{Y}$ が**実現可能** (*realizable*) とは，f の実現可能な具現化 $\mathtt{F} :\subseteq \mathbf{A} \to \mathbf{A}$ が存在することを意味する．関数 $f : \mathcal{X} \to \mathcal{Y}$ が**計算可能** (*computable*) であるとは，f の計算可能な具現化 $\mathtt{F} :\subseteq \mathbf{A} \to \mathbf{A}$ が存在することを意味する．

注意. 具現化と実現可能関数の英名は realizer と realizable function であるが，本稿では 1 つの realize という単語に対して，具現と実現という 2 つの訳語を用いている．訳語は統一すべきかもしれないが，具現化と実現可能関数は定義上は大きく異なるものであり，混乱を招き得るので，あえて訳語をずらした．

例 4.5. 例 4.2 で定義した $\mathbf{A}$-表現空間 $(\mathbb{N}, \nu_{\mathbb{N}})$ において，$n \mapsto n+1$ の具現化は $\underline{n} \mapsto \underline{n+1}$ であり，これは命題 3.35 より $\mathtt{succ} \in \mathbf{A}$ によって実現可能である．つまり，$f : (\mathbb{N}, \nu_{\mathbb{N}}) \to (\mathbb{N}, \nu_{\mathbb{N}})$ を $f(n) = n+1$ で定義すれば，この f は実現可能関数である．

相対部分組合せ代数 $\mathbf{A}$ に対し，しばしば $\mathsf{Mod}(\mathbf{A})$ によって $\mathbf{A}$-表現空間と計算可能関数のなす圏を表すことがある．部分組合せ代数による表現を考えるメリットは，部分組合せ代数が関数適用の抽象化であり，任意の $\mathbf{A}$-表現空間 X, Y に対して，関数空間 $[X \to Y]$ もまた $\mathbf{A}$-表現空間になるということである．専門用語を用いれば，$\mathsf{Mod}(\mathbf{A})$ は**デカルト閉圏** (*cartesian closed category*) をなすことが分かる．具体的には，以下のようにして関数空間は表現される．

定義 4.6. $\mathbf{A}$-表現空間 X, Y に対し，X から Y への $\mathbf{A}$-実現可能関数全体の空間は次の関数 $\mathtt{f} \mapsto [\![\mathtt{f}]\!]_{X \to Y}$ によって表現できる．

$$[\![\mathtt{f}]\!]_{X \to Y} = f \iff (\forall a)\ [\mathtt{f}a]_Y = f([a]_X).$$

つまり，関数 $f : X \to Y$ の名とは，f の具現化を実現する元 $\mathtt{f} \in \mathbf{A}$ である．この表現空間を $[X \to Y]_{\mathbf{A}}$ と書く．$\mathbf{A}$ が文脈から明らかな場合は，単に $[X \to Y]$ と書く．

事実，1950 年代に，クリーネは高階関数の計算理論を作り上げた．現代的な視点からは，クリーネの高階関数論の基本的な部分は，クリーネの第一または相対第二代数による表現空間の理論の一部として展開することができる．部分組合せ代数によって空間を表現することの有り難みは，その空間上の計算論を導入できるというだけでなく，様々な圏論的構成を受容できる，ということでもある．しかし，本稿では圏論についてはこれ以上深入りしない．

4.1 ナンバリングの理論

最も基本的な表現空間の理論は，クリーネの第一代数によるものである．命題 3.22 ではクリーネの第一代数を $\mathbb{K}_1$ と書いたが，クリーネの第一代数が単に自然数上の標準的な計算論であることを強調するため，以後，$\mathbf{N}$ によってクリーネの第一代数を表す．伝統的に，$\mathbf{N}$-表現は**部分ナンバリング** (*partial numbering*) と呼ばれ，全域 $\mathbf{N}$-表現は**ナンバリング** (*numbering*) と呼ばれる．また，$\mathbf{N}$-表現空間は**数化集合** (*numbered set*) と呼ばれることもある．しかし，本稿では用語の統一のために，これらの古典的な用語は用いない．この節では，クリーネの第一代数に基づく $\mathsf{Mod}(\mathbf{N})$ 上の計算論を概観しよう．つまり，自然数上の計算論に基づく理論によって，如何なる計算論が展開可能かを見る．

まず，命題 3.33 を用いると，任意の部分組合せ代数の中で有限列が 1 つの要素としてコードできる．たとえば 3 つの元 $a, b, c \in \mathbf{N}$ は，$\langle a, b, c\rangle = \langle\langle a, b\rangle, c\rangle \in \mathbf{N}$ のようにコードされる．

例 4.7 (有理数)**.** 各 $a, b \in \mathbb{N}$, $b \neq 0$ について，

$$\nu_{\mathbb{Q}}(\langle 0, a, b\rangle) = \frac{a}{b}, \qquad \nu_{\mathbb{Q}}(\langle 1, a, b\rangle) = -\frac{a}{b}$$

によって $\nu_{\mathbb{Q}} :\subseteq \mathbf{N} \to \mathbb{Q}$ を定義する．このとき，$(\mathbb{Q}, \nu_{\mathbb{Q}})$ は $\mathbf{N}$-表現空間である．

例 4.8 (計算可能関数の空間)**.** チューリング機械 M のコード e が与えられたとき，$[\![e]\!]$ によって，M によって計算される $\mathbb{N}$ 上の部分関数を表すものとする．このとき，$[\mathbb{N} \to \mathbb{N}_\perp]$ を $\mathbb{N}$ 上の部分計算可能関数全体の集合とすると，$([\mathbb{N} \to \mathbb{N}_\perp], [\![\cdot]\!])$ は $\mathbf{N}$-表現空間をなす．

例 4.9 (計算可枚挙集合の空間)**.** CE を $\mathbb{N}$ の計算可枚挙部分集合全体の集合とする．各 $e \in \mathbb{N}$ に対して，W_e を次のように定義する．

$$W_e = \{n \in \mathbb{N} : [\![e]\!](n) \downarrow\}.$$

このとき，$W : e \mapsto W_e$ に対して，(CE, W) は $\mathbf{N}$-表現空間をなす．

命題 3.22 の証明の直後で触れたように，クリーネの第一代数 $\mathbf{N} = (\mathbb{N}, \cdot)$ の代わりに $\Sigma^* = (\Sigma^*, \cdot)$ を考えてもよい．次の例では，Σ^*-表示空間を考えたほうが少し都合がよい．しかし，Σ^*-表示空間

と **N**-表示空間は計算可能性理論としては実質的に同一である．したがって，その多少の差異を気に留める必要はない．

例 4.10 (有限表示モノイド)**.** アルファベット Σ 上の関係 $R \subseteq \Sigma^* \times \Sigma^*$ が与えられたとき，各 $\sigma \in \Sigma^*$ に対して，

$$[\sigma]_R = \{\tau \in \Sigma^* : \sigma \equiv_R \tau\}$$

と定義する．ここで，$\equiv_R$ は例 1.19 で定義した自由モノイド Σ^* 上の合同関係である．このとき，$\sigma \mapsto [\sigma]_R$ は $\langle \Sigma \mid R \rangle$ によって与えられる商モノイド M_R の Σ^*-表現関数である．よって，$(M_R, [\cdot]_R)$ は Σ^*-表現空間をなす．

ここで，M_R の元のコードは一意ではないことに注意する．つまり，$x = [\sigma]_R \in M_R$ に対して，$\tau \equiv_R \sigma$ なる τ は全て x のコードである．

例 4.11 (有限有向グラフ)**.** $G = (V, E)$ を有限有向グラフとする．このとき，V と E は共に有限集合なので，$V = \{v(i)\}_{i \leqslant n}$ かつ $E = \{(v(a_i), v(b_i)) : i \leqslant k\}$ のように並べられるはずである．すると，定理 1.28 の証明中の文字列書換系と同様の方法で $G = (V, E)$ をコードできる:

$$\langle\langle a_0, b_0\rangle, \langle a_1, b_1\rangle, \langle a_2, b_2\rangle, \ldots, \langle a_k, b_k\rangle\rangle.$$

これによって，グラフ G の同型類を表しているものと考える．つまり，上のような有限有向グラフのコード $\sigma \in \Sigma^*$ が与えられたとき，$\mathrm{Gr}(\sigma)$ を σ によって表されるグラフとしたとき，

$$\sigma \simeq_{\mathrm{Graph}} \tau \iff \mathrm{Gr}(\sigma) \text{ と } \mathrm{Gr}(\tau) \text{ が有向グラフとして同型である}$$

と定義する．このとき，C_{Graphs} を有限有向グラフのコードを表している自然数全体とする，つまり $\mathrm{dom}(\mathrm{Gr})$ のこととする．$\mathsf{FinGraphs} = C_{\mathrm{Graphs}}/\simeq_{\mathrm{Graph}}$ を**有限有向グラフの同型類の N-表現空間**と呼び，**N**-表現関数は $\sigma \mapsto [\sigma]_{\mathrm{Graph}} = \{\tau : \tau \simeq_{\mathrm{Graph}} \sigma\}$ によって与えられる．

注意. 文字列書換系 $(\Sigma^*, \to)$ も有向グラフと考えられるが，これは無限グラフであることに注意する．また，例 4.10 の空間を無限グラフとして見たとき，これはあくまで 1 つの無限グラフを固定した上での連結成分全体の表現空間となり，例 4.11 のような有限グラフの同型類全体の表現空間とは全く異なるものである．

例 4.12 (遺伝的有限集合)**.** 集合 A が与えられたとき，$G_A = (\{A\}, \ni)$ を有向グラフだと思うこととする．A が**遺伝的有限集合** (*hereditarily finite set*) であるとは，G_A が有限グラフであることを意味する．したがって，例 4.11 と同じ方法で，全ての遺伝的有限集合をコードできる．つまり，有限有向グラフの同型類の **N**-表現空間 $\mathsf{FinGraphs}$ の部分空間として，遺伝的有限集合全体 $\mathbb{HF}$ は **N**-表現空間をなす．

定義 4.13. $\mathbf{A}$-表現空間 X が**決定可能** (*decidable*) または**計算離散** (*computably discrete*) であるとは，X 上の等号 $=$ が計算可能であることを意味する．言い換えれば，

$$\{(u,v) \in \mathbf{A} \times \mathbf{A} : \nu_X(u) = \nu_X(v)\}$$

が計算可能であることを言う．

例 4.14. 有限表示モノイド $\langle \Sigma \mid R \rangle$ の**語の問題** (*word problem*) とは，以下の集合

$$\mathsf{WP}_{\langle \Sigma | R \rangle} = \{(u,v) \in \Sigma^* \times \Sigma^* : u \equiv_R v\}$$

の要素関係の決定問題である．言い換えれば，与えられた $(u,v) \in \Sigma^*$ について，$(u,v) \in \mathsf{WP}_{\langle \Sigma | R \rangle}$ かどうかを判定する問題である．有限表示モノイド $\langle \Sigma \mid R \rangle$ の語の問題が**計算可能** (*computable*) または**可解** (*solvable*) であるとは，この判定が計算可能であることである．つまり，$\mathsf{WP}_{\langle \Sigma | R \rangle}$ が計算可能集合であることを意味する．

例 4.10 のように有限表示モノイド $\langle \Sigma \mid R \rangle$ を Σ^*-表現空間として見たとき，$\langle \Sigma \mid R \rangle$ が決定可能表現空間であることと語の問題 $\mathsf{WP}_{\langle \Sigma | R \rangle}$ が計算可能であることは同値である．

例 4.15. 例 4.11 の有限有向グラフの同型類の表示空間 $\mathsf{FinGraphs}$ および例 4.12 の遺伝的有限集合全体の表示空間 $\mathbb{HF}$ は決定可能空間である．なぜなら，与えられた 2 つの有限構造の間に全単射は有限種類しか存在しないので，それぞれが同型写像になっているかを順々に検証していけばよい．

さて，ナンバリングの理論の主題は，異なるナンバリングの比較である．チューリング完全な計算モデルが与えられれば，それは部分計算可能関数全体のナンバリング，つまり関数空間 $[\mathbb{N} \to \mathbb{N}_\perp]$ の $\mathbf{N}$-表現を与えるだろう．しかし，当然ながら，計算モデル毎に異なる $\mathbf{N}$-表現が作られる．重要な問題は，複数の計算モデル間の翻訳の問題である．我々が用いている計算モデル M によって，別人物が用いている計算モデル N における計算をシミュレートしたい．計算モデルを具体的なプログラミング言語に置き換えると，状況が分かりやすい．たとえば，M は Haskell で N は Java であるとすれば，Java のソースコードを Haskell のソースコードに変換するトランスコンパイラを作れるか，という問題となる．

数学的にこの翻訳問題を定式化しよう．$[\![e]\!]_M$ によって，計算モデル M においてコード e のプログラムが与える部分計算可能関数を表す．$[\![e]\!]_N$ と書いたときも同様である．$[\![\cdot]\!]_N$ から $[\![\cdot]\!]_M$ への翻訳問題は，次のような計算可能関数 $t : \mathbb{N} \to \mathbb{N}$ が存在するかどうかを尋ねるものである．

$$(\forall e \in \mathbb{N})\ [\![e]\!]_N \simeq [\![t(e)]\!]_M.$$

本稿では深入りしないが，ナンバリングの理論における最初の発見のひとつは，チューリング完全（すなわち全ての部分計算可能関数を実装できる）というだけでは，「万能機械」とは言い切れない，ということである．実際，互いに翻訳の存在しないチューリング完全な計算モデルが有り得る．これは部分組合せ代数とならないチューリング完全な計算モデルがあるということでもある．

以上は関数空間 $[\mathbb{N} \to \mathbb{N}_\perp]$ の **N**-表現に関する議論であるが，一般的な **A**-表現空間における翻訳概念を定義できる．

定義 4.16. ν_0, ν_1 を集合 X の **A**-表現とする．ν_0 が ν_1 に**還元可能** (*reducible*) であるとは，ある計算可能関数 $t :\subseteq \mathbf{A} \to \mathbf{A}$ が存在して，次が成立することを指す．

$$(\forall a \in \mathrm{dom}(\nu_0))\ \nu_0(a) = \nu_1(t(a)).$$

このとき，$\nu_0 \leqslant \nu_1$ と書く．また，$\nu_0 \leqslant \nu_1$ かつ $\nu_1 \leqslant \nu_0$ であるとき，$\nu_0 \equiv \nu_1$ と書く．

もし $\nu_0 \equiv \nu_1$ であれば, 同値な表現であると思って良い．集合 X の表現 ν_0, ν_1 について, $\nu_0 \leqslant \nu_1$ ということは，恒等写像 $\mathrm{id} : (X, \nu_0) \to (X, \nu_1)$ が計算可能ということと同値である．

同じ集合に対して，複数の同値でない表現を与えて，それらを同時に取り扱うことが多々ある．これは，同一の集合に対して複数の異なる位相構造や代数構造などを入れることと同じようなものである．集合の表現を与えるというのは，集合に表現を介して計算可能性構造を入れるということであり，そして計算可能性構造の入れ方は一意ではない．

たとえば，2 元集合 $\{0,1\}$ の表現を幾つか見てみよう．

例 4.17 (離散 2 点空間)**.** 恒等関数の $\{0,1\}$ への制限 $\mathrm{id}_2 : \{0,1\} \to \{0,1\}$ は $\{0,1\}$ の **N**-表現である．また，$\nu_2 : \mathbb{N} \to \{0,1\}$ を n が偶数ならば $\nu_2(n) = 0$ とし，n が奇数なら $\nu_2(n) = 1$ と定義すれば，$\nu_2 : \mathbb{N} \to \{0,1\}$ は $\{0,1\}$ の全域 **N**-表現である．**N**-表現空間 $(\{0,1\}, \mathrm{id}_2)$ と $(\{0,1\}, \nu_2)$ を離散 2 点空間と呼び，それぞれ $\mathbf{2}$ および $\mathbf{2}_\nu$ と書く．

例 4.18 (シエルピンスキ空間)**.** 2 元集合 $\{0,1\}$ に次の表現を与えたものを**シエルピンスキ空間** (*Sierpiński space*) と呼び，$\mathbb{S}$ と書く．

$$[e]_{\mathbb{S}} = \begin{cases} 1 & \text{if } [\![e]\!](0)\downarrow \\ 0 & \text{otherwise} \end{cases}$$

還元可能性証明の練習のために，以下を確認しよう．

命題 4.19. $\mathrm{id}_2 \equiv \nu_2 < [\cdot]_{\mathbb{S}}$.

Proof. $\mathrm{id}_2 \leqslant \nu_2$ であることは，任意の $i \in \{0,1\}$ について $\mathrm{id}_2(i) = \nu_2(i)$ であることから従う．$\nu_2 \leqslant \mathrm{id}_2$ について，$\nu_2 : \mathbb{N} \to \{0,1\}$ は自然数上の関数として計算可能であり，$t = \nu_2$ とすれば，任意の $n \in \mathbb{N}$ について $\nu_2(n) = \mathrm{id}_2(t(n))$ が成立する．続いて $\mathrm{id}_2 \leqslant [\cdot]_{\mathbb{S}}$ であるが，0 を入力した時に停止するプログラム e_0 と停止しないプログラム e_1 を作る．つまり，$[\![e_0]\!](0)\uparrow$ かつ $[\![e_1]\!](0)\uparrow$ であり，よって，$[e_0]_{\mathbb{S}} = 0$ かつ $[e_1]_{\mathbb{S}} = 1$ である．このとき $i \mapsto e_i$ は計算可能であり，$\mathrm{id}_2(i) = i = [e_i]_{\mathbb{S}}$ を得る．

最後に $[\cdot]_{\mathbb{S}} \not\leqslant \mathrm{id}_2$ を示す．もし $[\cdot]_{\mathbb{S}} \leqslant \mathrm{id}_2$ だとしたら，ある計算可能関数 $t : \mathbb{N} \to 2$ が存在して，任意の e について，$[e]_{\mathbb{S}} = t(e)$ となる．t は全域計算可能関数であるから，$\{e \in \mathbb{N} : t(e) = 1\}$ は計

算可能である．一方，$\{e \in \mathbb{N} : [e]_{\mathbb{S}} = 1\} = \{e \in \mathbb{N} : [\![e]\!](0)\downarrow\}$ となる．演習問題 2.4 より，これは計算不可能である．よって $[\cdot]_{\mathbb{S}} \not\leqslant \mathrm{id}_2$ を得る． □

これらの「表現が同値でない」ということについて「同じものを違う風に表現している」と常に考えるのは少しばかり誤解を生む．「同じものの表現が異なる」というよりは，「違うものだが集合部分がたまたま一致する」と考えた方が適切な状況もある．たとえば，離散 2 点空間 $\mathbf{2}$ および $\mathbf{2}_\nu$ とシエルピンスキ空間 $\mathbb{S}$ は，そもそも想定している空間が異なる．離散 2 点空間において，0 と 1 は平等である．一方，シエルピンスキ空間 $\mathbb{S}$ においては，0 と 1 は不平等である．

シエルピンスキ表現では，計算が「停止しない」という状況を 0 で表し，「停止する」という状況を 1 で表している．「停止しない」という状況から「停止する」という状況に計算が遷移することはあっても逆は有り得ない．この意味で，0 と 1 は異なる質を持っているのである．この 0 と 1 の不平等性，あるいは 0 から 1 への一方向遷移を明示した空間がシエルピンスキ空間である．

シエルピンスキ空間は計算可枚挙性と大きく関連する．これを説明するために，表現空間の同型性の概念を導入しよう．

定義 4.20. $\mathbf{A}$-表現空間 X, Y が**計算同型** (*computably isomorphic*) とは，ある全単射 $f : X \to Y$ で，f と f^{-1} が共に計算可能であるものが存在することを意味する．

定義 4.6 を用いて，関数空間 $[\mathbb{N} \to \mathbb{S}]$ を考えよう．我々の部分組合せ代数は $\mathbf{N}$ であるから，$[\mathbb{N} \to \mathbb{S}]$ は $\mathbb{N}$ から $\mathbb{S}$ への計算可能関数全体の空間であり，次の表現関数 $[\![\cdot]\!]_{\mathbb{N}\to\mathbb{S}}$ によって表現されている．

$$[\![\mathtt{f}]\!]_{\mathbb{N}\to\mathbb{S}} = f \iff (\forall n \in \mathbb{N})\ [[\![\mathtt{f}]\!](n)]_{\mathbb{S}} = f(n).$$

実は，この関数空間 $[\mathbb{N} \to \mathbb{S}]$ は，例 4.9 の計算可枚挙集合の空間 CE と同型である．

命題 4.21. $\mathbf{N}$-表現空間 $[\mathbb{N} \to \mathbb{S}]$ と CE は計算同型である．

Proof. 関数 $\Phi : [\mathbb{N} \to \mathbb{S}] \to \mathsf{CE}$ を次によって定義する．各 $g : [\mathbb{N} \to \mathbb{S}]$ に対して，

$$\Phi(g) = g^{-1}\{1\} = \{n \in \mathbb{N} : g(n) = 1\}.$$

この関数 Φ が計算同型写像であることを示そう．明らかに Φ は単射である．Φ の計算可能性について，$[\![p(e)]\!](n) \simeq [\![[\![e]\!](n)]\!](0)$ なる計算可能関数 $p : \mathbb{N} \to \mathbb{N}$ を取ると，任意の $e \in \mathbb{N}$ について，

$$\begin{aligned}\Phi([\![e]\!]_{\mathbb{N}\to\mathbb{S}}) = \{n \in \mathbb{N} : [[\![e]\!](n)]_{\mathbb{S}} = 1\} &= \{n \in \mathbb{N} : [\![[\![e]\!](n)]\!](0)\downarrow\} \\ &= \{n \in \mathbb{N} : [\![p(e)]\!](n)\downarrow\} = W_{p(e)}\end{aligned}$$

となるから，p は Φ を具現化する．一方，$[\![e]\!](n) \simeq [\![[\![q(e)]\!](n)]\!](0)$ なる計算可能関数 $q : \mathbb{N} \to \mathbb{N}$ を取ると，

$$W_e = \{n \in \mathbb{N} : [\![e]\!](n)\downarrow\} = \{n \in \mathbb{N} : [\![[\![q(e)]\!](n)]\!](0)\downarrow\} = \Phi([\![q(e)]\!]_{\mathbb{N}\to\mathbb{S}})$$

であるから，q は Φ^{-1} を具現化する．以上より，Φ が計算同型写像であることが示された． □

4.2 ライスの定理と空間の連結性

本節では, 1953 年にヘンリー・ライス (Henry G. Rice) によって証明された**ライスの定理** (*Rice's theorem*) を取り扱う．これは，次のような驚くべき主張をする定理である．

> 部分計算可能関数に関する非自明な性質 P が任意に与えられている．このとき，与えられたプログラム e の計算する関数が P を満たすか否かを判定するアルゴリズムは存在しない．

第 2 節では，個々の決定問題が計算可能か計算不可能かを議論してきたが，ライスの定理は一度で大量の決定問題の計算不可能性を導く．究極の計算不可能性定理の 1 つである．

ライスの定理は，一見すると非常に興味深いが，しかし，クリーネの再帰定理などとは異なり，定理自体の有用性は低く，理論的にも深みがあるわけではない．それにも関わらず，ライスの定理は何故か伝統的に計算可能性理論入門における必須トピックとして取り扱われる．これは，その定理の衝撃的な内容に比較すると，証明が極めて簡単であるからであろう．

しかし，それだけでは，必須トピックとして扱うには少し説得力が足りない．ここでは，ライスの定理の空間的側面，つまり，これが計算可能性理論の概念を空間的に理解するためには良いトピックであることを強調する．「計算と空間」というコンテキストに置くことで，ライスの定理は特筆すべき価値を持つ．本節における我々の目標は，ライスの定理に対する以下のような幾何学的イメージを持つことである．

> 自然数や文字列の世界は離散空間（デジタル）であるが，自然数や文字列上の部分計算可能関数全体を空間として見ると，そこは連結空間のようになっている．

定義 4.1 において，表現空間 X の要素 x のコード全体の集合を $||x||_X$ と書いていたことを思い出そう．X が $\mathbf{N}$-表現空間の場合，部分集合 $S \subseteq X$ について，S の元のコード全体の集合 $||S||_X = \bigcup_{x \in S} ||x||_X$ は伝統的に**添字集合** (*index set*) と呼ばれることがある．

定義 4.22. X を全域 $\mathbf{N}$-表現空間とする．集合 $A \subseteq X$ が**計算可能開** (*computably open*) とは，添字集合 $||A||_X$ が計算可枚挙であることを意味する．

この概念は単に計算可枚挙と呼ばれることも多いが，本稿では，空間としての側面を強調するために，これを計算可能開集合と呼ぶ．計算可能開集合 A とは，与えられたコード e が A の要素を表すかどうかに関する半計算可能な手続きがあるということである．計算可能開集合全体を開基とする全域 $\mathbf{N}$-表現空間上の位相は**エルショフ位相** (*Ershov topology*) と呼ばれる．

例 4.23. 例 4.17 の離散 2 点空間 $\mathbf{2}$ において，$\varnothing, \{0\}, \{1\}, \{0,1\}$ は全て計算可能開集合である．一方，例 4.18 のシエルピンスキ空間 $\mathbb{S}$ においては，$\varnothing, \{1\}, \{0,1\}$ は計算可能開集合だが，$\{0\}$ は計算可能開集合ではない．

豆知識. シエルピンスキ空間上のエルショフ位相を考えると，位相空間論におけるシエルピンスキ空間と同相になる．

計算可枚挙集合を開集合として見ると，計算可能関数は連続関数に相当する．位相空間論において，連続関数とは，開集合の逆像が開集合となるような関数であった．

命題 4.24. X と Y を全域 **N**-表現空間であり，$f : X \to Y$ を計算可能関数とする．このとき，任意の計算可能開集合 $U \subseteq Y$ に対して，$f^{-1}[U]$ も X の計算可能開集合である．

Proof. $f : X \to Y$ が計算可能関数であれば，f を具現化する計算可能関数 $\mathtt{f} : \mathbb{N} \to \mathbb{N}$ が存在する．ν_X と ν_Y をそれぞれ X と Y の表現とすれば，つまり，$f \circ \nu_X = \nu_Y \circ \mathtt{f}$ である．このとき，$f^{-1}[U]$ のコードの集合は以下によって与えられている．

$$\begin{aligned}||f^{-1}[U]||_X = ||\{x \in X : f(x) \in U\}||_X =& \{n \in \mathbb{N} : f(\nu_X(n)) \in U\} \\ =& \{n \in \mathbb{N} : \nu_Y(\mathtt{f}(n)) \in U\} = \{n \in \mathbb{N} : \mathtt{f}(n) \in ||U||_Y\}.\end{aligned}$$

仮定より，U は計算可能開集合であるから，$||U||_Y$ は計算可枚挙である．$\mathtt{f}$ は計算可能であるから，明らかに $\{n \in \mathbb{N} : \mathtt{f}(n) \in ||U||_Y\}$ も計算可枚挙である．よって，$||f^{-1}[U]||_X$ は計算可枚挙であり，つまり $f^{-1}[U]$ が計算可能開集合であることが示された． □

豆知識. 命題 4.24 の性質を強めて，計算可能開集合 U のコードから計算可能開集合 $f^{-1}[U]$ のコードを返す計算可能関数が存在する，という主張にすると，これが f の計算可能性と同値になる．

定義 4.25. 全域 **N**-表現空間 X が**計算連結** (*computably connected*) とは，X の非自明な計算可能直和分解が存在しないことである．つまり，$X = X_0 \cup X_1$ かつ $X_0 \cap X_1 = \varnothing$ となるような空でない計算可能開部分集合 $X_0, X_1 \subseteq X$ が存在しないことを意味する．

計算連結性はライスの定理と深く関わっている．実際，以下は容易に確かめられる．

$$\text{計算連結} \iff \text{自明なものを除く添字集合が全て計算不可能.}$$

なぜなら，計算連結性は添字集合 $||X_0||_X$ と $||X_1||_X$ が共に計算可枚挙であることを述べるが，X_0 と X_1 は全体空間 X の分割なので，$||X_1||_X = \mathbb{N} \backslash ||X_0||_X$ である．よって，ポストの定理 (系 3.14) より，これは添字集合 $||X_0||_X$ と $||X_1||_X$ が共に計算可能であることと同値である．
したがって，ライスの定理とは，空間 $[\mathbb{N} \to \mathbb{N}_\perp]$ の計算連結性を述べる定理である．

例 4.26. シエルピンスキ空間 $\mathbb{S}$ は計算連結である．なぜなら，$\mathbb{S}$ の非自明な分割は $\mathbb{S} = \{0\} \cup \{1\}$ のみであるが，$\{0\}$ の添字集合は $||0||_{\mathbb{S}} = \{e \in \mathbb{N} : [e](0) \uparrow\}$ であり，これは計算可枚挙ではない．

ところで定義 4.13 において，計算離散性という概念を導入した．2 点以上を含む離散空間は不連結であるという位相空間論的事実は，計算可能性理論では次のように表される．

例 4.27. 2 点以上を含む **N**-表現空間は，もし計算離散ならば計算連結ではない．

定義 4.28. $(X, [\cdot]_X)$ を **N**-表現空間とする．このとき，X の**持ち上げ** (*lifting*) とは，集合 $X_\perp = X \cup \{\perp_X\}$ に次の表現 $[\cdot]_{X_\perp}$ を与えた **N**-表現空間である．

$$[e]_{X_\perp} = \begin{cases} [[\![e]\!](0)]_X & \text{if } [\![e]\!](0)\downarrow \\ \perp_X & \text{otherwise.} \end{cases}$$

例 4.29. 自然数全体の集合 $\mathbb{N}$ は，恒等写像によって表現することにより，自明に **N**-表現空間となる．このとき，$\mathbb{N}_\perp$ の表現は

$$[e]_{\mathbb{N}_\perp} = \begin{cases} [\![e]\!](0) & \text{if } [\![e]\!](0)\downarrow \\ \perp_{\mathbb{N}} & \text{otherwise.} \end{cases}$$

によって与えられる．このような空間は**平坦領域** (*flat domain*) と呼ばれる．

シエルピンスキ空間は一点空間 $\mathbf{1} = \{\bullet\}$ の持ち上げと考えることもできる．持ち上げに関する重要な性質として，必ずレトラクション $r : X_{\perp\perp} \to X_\perp$ が存在する，ということがある．ここで，集合 X の部分集合 $S \subseteq X$ に対して，**レトラクション** (*retraction*) とは関数 $r : X \to S$ で，$r \restriction S$ が恒等関数であるようなもののことである．つまり，任意の $x \in S$ に対して，$r(x) = x$ を満たす．

この性質を $X_\perp$ のような $\perp$ を持つ空間の本質であると考えよう．このアイデアを元に，$\perp$ に相当するものが既に存在している空間を次のように抽象化する．

定義 4.30. 全域 **N**-表現空間 Z が **$\perp$-レトラクト** ($\perp$-*retract*) であるとは，計算可能レトラクション $r_Z : Z_\perp \to Z$ が存在することを意味する．

注意. $\perp$-レトラクトは**フォーカル集合** (*focal set*) とも呼ばれる．

演習問題 4.31. 任意の全域 **N**-表現空間 X に対して，$X_\perp$ は $\perp$-レトラクトであることを示せ．

例 4.32. $\mathbb{N}$ 上の部分計算可能関数の空間 $[\mathbb{N} \to \mathbb{N}_\perp]$ は $\perp$-レトラクトである．ここで，$[\mathbb{N} \to \mathbb{N}_\perp]$ は $[\![\cdot]\!]$ によって表現されている．関数空間 $[\mathbb{N} \to \mathbb{N}_\perp]$ の持ち上げ $[\mathbb{N} \to \mathbb{N}_\perp]_\perp$ の表現を $[\![\cdot]\!]_\perp$ と書くことにする．次のような r が計算可能であることを示せばよい．任意の $f \in [\mathbb{N} \to \mathbb{N}_\perp]_\perp$ に対して，

$$r(f) = \begin{cases} f & \text{if } f \in [\mathbb{N} \to \mathbb{N}_\perp], \\ \varnothing & \text{if } f \notin [\mathbb{N} \to \mathbb{N}_\perp]. \end{cases}$$

ここで，$\varnothing$ は至る所未定義な関数を表す．このような関数 r は明らかにレトラクションである．

r の計算可能性を示すため，$d(e)$ を次のようなチューリング機械のコードとする．入力 n に対して，プログラム $[\![e]\!](0)$ が停止するのを待ち，停止したら $[\![[\![e]\!](0)]\!](n)$ を実行する．

$$[\![e]\!](0)\downarrow \implies (\forall n)\ [\![d(e)]\!](n) \simeq [\![[\![e]\!](0)]\!](n) \implies [\![d(e)]\!] \simeq [\![[\![e]\!](0)]\!] \simeq [\![e]\!]_\perp$$
$$[\![e]\!](0)\uparrow \implies (\forall n)\ [\![d(e)]\!](n)\uparrow \implies [\![d(e)]\!] = \varnothing.$$

したがって，計算可能関数 $d : \mathbb{N} \to \mathbb{N}$ は明らかに r の具現化である．

例 4.33. 例 4.32 の議論を少し修正すると，任意の全域 **N**-表現空間 X, Y について，X から Y への部分関数全体の空間 $[X \to Y_\perp]$ は $\perp$-レトラクトであることが示される．特に，命題 4.21 より，$\mathbb{N}$ 上の計算可枚挙集合の空間 $\mathsf{CE} \simeq [\mathbb{N} \to \mathbb{S}] \simeq [\mathbb{N} \to \mathbb{S}_\perp]$ は $\perp$-レトラクトである．

定理 4.34 (抽象ライスの定理)**.** 任意の $\perp$-レトラクトは計算連結である．

Proof. X を $\perp$-レトラクトとする．まず，次を示す．

主張. 任意の $x \in X$ に対して，$f_x : \mathbb{S} \to X_\perp$ を $f_x(1) = x$ かつ $f_x(0) = \perp_X$ で定義すると，f_x は計算可能である．

これを確かめるためには，f の具現化 $\mathtt{f} : \mathbb{N} \to \mathbb{N}$ が計算可能であることを示せばよい．この $\mathtt{f}$ は次の性質を満たすはずである．入力 $e \in \mathbb{N}$ に対して，もし $[\![e]\!](0)$ が停止するならば x の $X_\perp$-コードを出力し，停止しないならば $\perp_X$ の $X_\perp$-コードを出力する．

プログラム P_e を，どんな入力が与えられても，$[\![e]\!](0)$ が停止するのを待ってから x のコードを出力するものとしよう．入力 $e \in N$ に対して，プログラム P_e のコード $\mathtt{f}(e)$ を返す関数 $\mathtt{f}$ は容易に計算可能である．このとき，特に

$$[e]_{\mathbb{S}} = 1 \iff [\![e]\!](0)\downarrow \iff [\![\mathtt{f}(e)]\!](0)\downarrow \in ||x||_X \iff |[\![\mathtt{f}(e)]\!](0)|_X = x \iff [\mathtt{f}(e)]_{X_\perp} = x$$
$$[e]_{\mathbb{S}} = 0 \iff [\![e]\!](0)\uparrow \iff [\![\mathtt{f}(e)]\!](0)\uparrow \iff [\mathtt{f}(e)]_{X_\perp} = \perp_X$$

であるから，$\mathtt{f}$ は f_x の計算可能な具現化となっている． ⊣ (主張)

主張. 任意の計算可能開集合 $U \subseteq X$ について，

$$r_X(\perp_X) \in U \implies U = X.$$

$r_X : X_\perp \to X$ を計算可能レトラクトとする．このとき，$r_X(\perp_X) \in U \neq X$ なる計算可能開集合 U が存在すると仮定して矛盾を導こう．まず，$x \in X \backslash U$ を取る．このとき，$r_x^{-1}[U] = U \cup \{\perp_X\}$ である．いま，$h = r_X \circ f_x$ とすると，$h^{-1}[U] = \{0\}$ である．前の主張より，$h : \mathbb{S} \to X$ は計算可能である．U は X の計算可能開集合であるから，命題 4.24 より，$h^{-1}[U] = \{0\}$ も $\mathbb{S}$ の計算可能開集合である．しかし，例 4.23 で見たように，$\{0\}$ は $\mathbb{S}$ の計算可能開集合ではありえない．したがって，任意の計算可能開集合 U について，$r_X(\perp_X) \in U \neq X$ では有り得なかった，つまり $r_X(\perp_X) \in U$ ならば $U = X$ だったということである． ⊣ (主張)

さて，計算可能開集合 $X_0, X_1 \subseteq X$ について，もし $X = X_0 \cup X_1$ ならば，$r_X(\perp_X) \in X_0$ または $r_X(\perp_X) \in X_1$ である．よって，上の主張により，$X_0 = X$ または $X_1 = X$ を得る．以上より，X の計算連結性が示された． □

系 4.35 (ライスの定理)**.** 部分計算可能関数の計算可能な添字集合は自明なものだけである．つまり，$P \subseteq [\mathbb{N} \to \mathbb{N}_\perp]$ が $P \neq \varnothing$ または $P \neq [\mathbb{N} \to \mathbb{N}_\perp]$ であるものとすれば，

$$\{e \in \mathbb{N} : [\![e]\!] \in P\}$$

は計算不可能である．

Proof. 例 4.32 で見たように，$[\mathbb{N} \to \mathbb{N}_\perp]$ は $\perp$-レトラクトであるから，定理 4.34 より計算連結である．計算連結性は，非自明な添字集合が全て計算不可能であるということと同値であったから，目的の主張は示される． □

ここで，例 4.33 より，自然数上の部分計算可能関数に関する添字集合だけでなく，任意の全域 $\mathbf{N}$-表現空間上の部分計算可能関数に関する添字集合に対して，計算可能なものは自明なものしかないことが示される．たとえば，命題 4.21 を用いれば，計算可枚挙集合の空間 CE は関数空間 $[\mathbb{N} \to \mathbb{S}]$ として表すことができるので，CE においてもライスの定理が成立する．

例 4.36. ライスの定理より，以下の集合は全て計算不可能である．

$$\begin{aligned}
\mathsf{Halt}_0 &= \{e \in \mathbb{N} : [\![e]\!](0)\downarrow\},\\
\mathsf{Fin} &= \{e \in \mathbb{N} : W_e \text{ is finite}\},\\
\mathsf{Inf} &= \{e \in \mathbb{N} : W_e \text{ is infinite}\},\\
\mathsf{Tot} &= \{e \in \mathbb{N} : (\forall n \in \mathbb{N})\ [\![e]\!](n)\downarrow\},\\
\mathsf{Cof} &= \{e \in \mathbb{N} : \mathbb{N}\backslash W_e \text{ is finite}\},\\
\mathsf{Rec} &= \{e \in \mathbb{N} : W_e \text{ is computable}\}.
\end{aligned}$$

後に確認するが，これらの添字集合の計算不可能性の度合いについて，以下が成立する．

$$\varnothing' \equiv_T \mathsf{Halt}_0 <_T \varnothing'' \equiv_T \mathsf{Fin} \equiv_T \mathsf{Inf} \equiv_T \mathsf{Tot} <_T \varnothing''' \equiv_T \mathsf{Cof} \equiv_T \mathsf{Rec}.$$

ここで，第 3.1 節で定義したように，$\leqslant_T$ はチューリング還元可能性であり，X' は X のチューリング・ジャンプである．つまり，X' は X-相対的計算に関する停止問題 Halt^X を表す．したがって，たとえば，$\varnothing''' = \mathsf{Halt}^{\mathsf{Halt}^{\mathsf{Halt}}}$ である．

注意. 位相空間論を既に学んでいる読者のために補足しておくと，抽象ライスの定理 4.34 の証明は，任意の $\perp$-レトラクトがエルショフ位相の下で（位相空間論の意味で）連結であることを述べている．なぜなら，エルショフ位相の定義より，任意の開集合は計算可能開集合を含むので，定理 4.34 の 2 つめの主張は，$r_X(\perp_X)$ を含む全ての開集合は空間全体に他ならないことを述べている．したがって，特に，ライスの定理 (系 4.35) とは，部分計算可能関数の空間 $[\mathbb{N} \to \mathbb{N}_\perp]$ がエルショフ位相の下で連結であることを主張する定理である．

ところで，唐突に現れた $\perp$-レトラクトというものの出自が気になる人もいるかもしれない．これは，ナンバリングの理論において，1960 年頃にマルツェフによって導入された**完全ナンバリング** (*complete numbering*) または**エルショフ完全ナンバリング** (*Ershov-complete numbering*) と呼ばれるものに相当する．エルショフは，この概念を $\mathbf{N}$-表現空間と計算可能関数の圏における単射的対象の類似物として説明している．ここでは，少々，単純化したバージョンを導入しよう．

定義 4.37. 全域 **N**-表現空間 Z が CE-**絶対拡張手** (*CE-absolute extensor*) とは，与えられた全域 **N**-表現空間 X と計算可能開集合 $S \subseteq X$ について，どんな計算可能関数 $f : S \to Z$ も全域計算可能関数 $\tilde{f} : X \to Z$ に拡張できることを意味する．

注意. トポロジーの意味での絶対拡張手は，**閉**部分集合上の連続関数の拡張可能性を考えるため，実際にはかなり異なる概念となることに注意する．

命題 4.38. 全域 **N**-表現空間が CE-絶対拡張手であることと $\perp$-レトラクトであることは同値である．

Proof. 全域 **N**-表現空間 Z が CE-絶対拡張手であるとする．このとき，持ち上げ $Z_\perp$ において $Z \subseteq Z_\perp$ の添字集合は半計算可能集合 $\{e : [\![e]\!](0)\downarrow\}$ であるから，Z は $Z_\perp$ の計算可能開部分集合である．恒等関数 $\mathrm{id} : Z \to Z$ は計算可能であるから，全域計算可能関数 $\tilde{\mathrm{id}} : Z_\perp \to Z$ に拡張される．これは明らかに計算可能レトラクションである．

逆に，Z が $\perp$-レトラクトであるとする．全域 **N**-表現空間 X とその計算可能開部分集合 S が与えられているとする．I_S を S の添字集合とすると，I_S は計算可枚挙であり，つまり半計算可能であるから，次のような計算可能関数 $p : \mathbb{N} \to \mathbb{N}$ が存在する．

$$p(n) = \begin{cases} n & \text{if } n \in I_S, \\ \uparrow & \text{if } n \notin I_S. \end{cases}$$

いま，計算可能関数 $f : S \to Z$ が与えられているとする．このとき $\mathtt{f} :\subseteq \mathbb{N} \to \mathbb{N}$ を f の計算可能な具現化とする．つまり，$n \in I_S$ について $f([n]_X) = [\mathtt{f}(n)]_Z$ を満たすものとする．このとき，$\mathtt{f} \circ p$ は計算可能であるから，そのコードを q とする．つまり，$[\![q]\!](n) \simeq \mathtt{f} \circ p(n)$ である．パラメータ定理より，ある計算可能関数 $t : \mathbb{N} \to \mathbb{N}$ が存在して，$[\![t(n)]\!](0) \simeq \mathtt{f} \circ p(n)$ となる．いま，

$$\begin{aligned} n \in I_S &\implies [\![t(n)]\!](0) \simeq \mathtt{f} \circ p(n) = \mathtt{f}(n)\downarrow \implies [t(n)]_{Z_\perp} = [\mathtt{f}(n)]_Z = f([n]_X) \\ n \notin I_S &\implies [\![t(n)]\!](0) \simeq \mathtt{f} \circ p(n)\uparrow \implies \perp_X \end{aligned}$$

であるから，t は全域計算可能関数 $g : X \to Z_\perp$ で $f \upharpoonright S = g \upharpoonright S$ となるものを具現化する．いま，$r_Z : Z_\perp \to Z$ を計算可能レトラクションとすれば，$\tilde{f} = r_Z \circ g : X \to Z$ は f を拡張する全域計算可能関数である． □

豆知識. 発展的なトピックとして，ライスの定理を強めた定理が幾つかある．1950 年代半ばに証明された**マイヒル-シェファードソンの定理** (*Myhill-Shepherdson theorem*) と**ライス-シャピロの定理** (*Rice-Shapiro theorem*) はよく知られている．$\mathbb{N}$ 上の有限関数 $\theta :\subseteq \mathbb{N} \to \mathbb{N}$ に対して，$[\theta]$ を θ を拡張する部分計算可能関数全体の集合とし，有限集合 $D \subseteq \mathbb{N}$ に対して，$[D]$ を D を含む $\mathbb{N}$ の部分集合全体の集合とする．つまり，

$$[\theta] = \{\psi \in [\mathbb{N} \to \mathbb{N}_\perp] : \theta \subseteq \psi\}, \qquad [D] = \{S \subseteq \mathbb{N} : D \subseteq S\}.$$

マイヒル-シェファードソンの定理は，集合族 $\{\varnothing\} \cup \{[\theta] : \theta$は有限関数 $\}$ が $[\mathbb{N} \to \mathbb{N}_\perp]$ 上のエルショフ位相の開基をなすことを述べる定理である．つまり，任意の空でない計算可能開集合 $U \subseteq [\mathbb{N} \to \mathbb{N}_\perp]$ に対して，ある有限関数の族 F が存在して，$U = \bigcup_{\theta \in F} [\theta]$ と書ける．同様に，ライス-シャピロの定理は，集合族 $\{\varnothing\} \cup \{D \subseteq \mathbb{N} : D$ は有限集合 $\}$ が CE 上のエルショフ位相の開基をなすことを述べる．つまり，任意の空でない計算可能開集合 $U \subseteq \mathsf{CE}$ に対して，ある有限集合の族 F が存在して，$U = \bigcup_{D \in F} [D]$ と書ける．

4.3 実数の計算論

計算理論の誕生以来，実数上の計算論は計算可能性理論の中心的テーマであり続けた．しかし，もちろん，あらゆる実数を有限文字列として取り扱うということは不可能である．それでは，研究者たちは如何にして実数上の計算論を取り扱ってきただろうか．1 つの解法は，クリーネの第一代数 $\mathbf{N}$ における計算可能実数論であり，もう 1 つの解法は，クリーネの相対第二代数 $\mathbf{K} := (\mathbb{K}_2, \mathbb{K}_{2\circ})$ における実数上の計算可能関数論である．実数の計算論の初期は前者のアプローチが主流だったように思うが，時代を経るにつれ，徐々に後者の理論の方が優れていると分かってきた．

この正確な意味を説明する前に，まず，実数の計算論の具体的なアイデアを述べよう．まず，有理数上の計算論は，例 4.7 のような有理数の表現を用いて自明に展開できる．実数の計算可能性を導入する最も簡単だが最も優れた方法は，任意精度計算（精度保証計算）である．

定義 4.39. 実数 $x \in \mathbb{R}$ が**任意精度計算可能**とは，任意の正有理数 ε 精度で x を有理近似するアルゴリズムが存在することである．つまり，ある計算可能関数 $\Phi : \mathbb{Q}_{>0} \to \mathbb{Q}$ が存在して，次を満たすことである．

$$(\forall \varepsilon \in \mathbb{Q}_{>0}) \quad |x - \Phi(\varepsilon)| < \varepsilon.$$

とりあえず，これが実数の計算可能性の妥当な定義のように思うが，少しだけ歴史的経緯を振り返ろう．チューリングが 1936 年にチューリング機械を導入した記念碑的論文「計算可能数とその決定問題への応用」では，実数の計算可能性は 2 進小数展開の計算可能性として導入されている．ここでは，その計算可能性を 2 進計算可能性と呼ぶことにする．

定義 4.40. 実数 $x \in \mathbb{R}$ が **2 進計算可能** (*binary computable*) とは，x の任意の 2 進小数展開

$$a_0 a_1 \dots a_n . a_{n+1} a_{n+2} \dots a_{n+k} a_{n+k+1} \dots$$

に対して，関数 $i \mapsto a_i$ が計算可能であることを意味する．

しかし，すぐにチューリングは，実数の計算論を 2 進展開で導入するのは誤りだと気づいたようである．翌年にチューリングは「計算可能数とその決定問題への応用 – 訂正」を出版し，その後半部では，自身の実数の 2 進計算論を撤回した．実数の計算論の正しい与え方として，チューリングは，たとえばブラウワーの直観主義数学における縮小区間による実数の表現を挙げている．これは区間の縮小列によって実数を表現するものである．

定義 4.41. 実数 $x \in \mathbb{R}$ が**区間計算可能**とは，有理数の対の計算可能な列 $(p_n, q_n)_{n\in\mathbb{N}}$ で次のようなものが存在することである．

$$(\forall n \in \mathbb{N})\ [p_n < p_{n+1} < q_{n+1} < q_n],\ \text{and}\ x = \lim_{n\to\infty} p_n = \lim_{n\to\infty} q_n.$$

任意精度計算との関連性に触れておくと，区間計算において，ストリーム $(p_n, q_n)_{n\in\mathbb{N}}$ の各段階では実数を両側から挟み込んでいるため，常に近似精度の保証が可能となっている．実際，以下のように，チャーチ・チューリングの提唱の実数の計算論版のようなものも成立する．

命題 4.42. 実数について，任意精度計算可能性，2 進計算可能性，区間計算可能性はいずれも同値である．

Proof. 任意精度計算可能性と区間計算可能性の同値性は読者の演習問題とする．任意の 2 進計算可能実数が任意精度計算可能であることを確認しよう．実数の与えられた無限 2 進小数表記 $\alpha = a_0a_1 \dots a_k.a_{k+1}a_{k+2} \dots$ に対して，q_n を有限小数 $a_0a_1 \dots a_k.a_{k+1}a_{k+2} \dots a_{k+n}a_{k+n+1}$ が表す有理数とする．このとき，$q = (q_n)_{n\in\omega}$ が $[\alpha]_{\text{bin}}$ の任意精度近似，つまり $[q]_A = [\alpha]_{\text{bin}}$ であることは容易に分かる．

最後に，任意精度計算可能実数が 2 進計算可能実数であることを示す．有理数がどちらの意味でも計算可能であることは明らかなので，無理数 x について任意精度計算可能性が 2 進計算可能性を導くことを示せばよい．また，$x \in [0, 1]$ であると仮定しても一般性を失わない．x を任意精度計算可能な無理数とし，Φ をその任意精度近似とする．x は無理数であるから，2 進小数展開した際に，必ず 0 と 1 の両方を無限に含む．有理数 $\Phi(s)$ を 2 進小数展開した結果を $\alpha_s = 0.a_0^s a_1^s \dots a_{\ell(s)}^s$ と書く．任意の $n \in \mathbb{N}$ について，α_s の小数点以下 $n+1$ 桁目以降 s 桁目以前に 01 または 10 が出現するような s を探す．このとき，$\Phi(s)$ から最大 2^{-s} の誤差が発生したとしても，この 01 または 10 の出現以前の部分は変動しない．よって，$a_n = a_n^s$ と定義すれば，$[0.a_1a_2 \dots]_{\text{bin}} = \lim_s \Phi(s) = x$ を得る． □

ところで，チューリングの最初の定義である「2 進計算可能性」とチューリングの第 2 の定義である「区間計算可能性」が同値であるならば，チューリングは訂正論文を出す必要は無かったのでは，と思うかもしれない．しかし，実は，チューリングが訂正論文を出した判断は正しかった．実関数の計算可能性を考える段階になると「2 進計算可能性」と「区間計算可能性」は大きく異なる．そして，そのとき「2 進計算可能性」は破綻する．なんと，実数を 3 倍するだけのごく単純な関数 $x \mapsto 3x$ ですら，2 進計算可能ではないのである．これについては，後の定理 4.45 で確認する．

このような問題を説明する前に，実数の表現について考えよう．そもそも実数の定義とは何であっただろうか．実数の集合 $\mathbb{R}$ とは，有理数の集合 $\mathbb{Q}$ の完備化，すなわちコーシー列の同値類であると学んだかもしれない．あるいは，実数とは，デデキント切断であると学んだかもしれない．

■コーシー実数: 有理**コーシー列** (*Cauchy sequence*) とは有理数列 $(q_n)_{n\in\omega} \in \mathbb{Q}^{\mathbb{N}}$ で,

$$(\forall n > 0)(\exists k \in \mathbb{N})(\forall i, j \geqslant k)\ |q_i - q_j| < 2^{-n}$$

を満たすものである．有理コーシー列 $(p_n)_{n\in\omega}$ と $(q_n)_{n\in\omega}$ が等しいとは，任意の $\varepsilon > 0$ について，十分大きな任意の $i, j \in \mathbb{N}$ について $|p_i - q_j| < \varepsilon$ が成立することである．実数の集合 $\mathbb{R}$ の**素朴コーシー表現** (*naive Cauchy representation*) とは，有理コーシー列の同値類として実数を表現する関数 $[\cdot]_{\mathrm{nC}} :\subseteq \mathbb{Q}^{\mathbb{N}} \to \mathbb{R}$ である．より正確には，任意の有理コーシー列 $(q_n)_{n\in\omega}$ について,

$$[(q_n)_{n\in\omega}]_{\mathrm{nC}} = \{(p_n)_{n\in\omega} : (p_n) \text{ は有理コーシー列であり，} (p_n) \text{ と } (q_n) \text{ は等しい.}\}$$

同値類 $[(q_n)_{n\in\omega}]_{\mathrm{nC}}$ のことを $\lim_{n\to\infty} q_n$ と表す．素朴コーシー表現は非常に使い勝手が悪い．本節で言及する実数の表現のうちでは最悪な表現である．コーシー列の収束速度が分からないので，コーシー列というストリームを読み込む過程で，極限の値にどれくらい近づいたか判断できないからである．素朴コーシー表現は極限概念を伴う表現であり，第 5 節で詳述するが，極限は容易に計算不可能性を生み出すのである．

そういうわけで，コーシー列と言った場合には，収束速度の情報が常に備わっていて欲しい．関数 $k : \mathbb{N} \to \mathbb{N}$ が列 $(q_n)_{n\in\omega}$ の**収束係数**とは,

$$(\forall n > 0)(\forall i, j \geqslant k(n))\ |q_i - q_j| < 2^{-n}$$

を満たすことを意味する．実数の集合 $\mathbb{R}$ の**係数付きコーシー表現**とは，有理コーシー列の正しい収束係数が与えられたとき，その有理コーシー列の同値類として実数を表現する関数 $[\cdot]_{\mathrm{mC}} :\subseteq \mathbb{Q}^{\mathbb{N}} \times \mathbb{N}^{\mathbb{N}} \to \mathbb{R}$ である．つまり，任意の有理コーシー列 $(q_n)_{n\in\omega}$ について，もし k が $(q_n)_{n\in\omega}$ の収束係数であるときのみ，$[(q_n), k]_{\mathrm{mC}}$ を定義し，$[(q_n), k]_{\mathrm{mC}} = \lim_{n\to\infty} q_n$ とする．

係数付きコーシー表現は非常に良い表現なのだが，定義が若干ごちゃごちゃするという難点がある．このため，係数付きコーシー表現の代わりに，id が収束係数となるようなコーシー列（急収束コーシー列）だけを考えて単純化することが多い．つまり，**急収束コーシー列** (*rapidly converging Cauchy sequence*) とは有理数列 $(q_n)_{n\in\omega} \in \mathbb{Q}^{\mathbb{N}}$ で,

$$(\forall n > 0)(\forall i, j \geqslant n)\ |q_i - q_j| < 2^{-n}$$

を満たすものである．このとき，実数の集合 $\mathbb{R}$ の**コーシー表現** (*Cauchy representation*) とは，急収束コーシー列の同値類として実数を表現する関数 $[\cdot]_{\mathrm{C}} :\subseteq \mathbb{Q}^{\mathbb{N}} \to \mathbb{R}$ である．つまり，$(q_n)_{n\in\omega}$ が急収束コーシー列であるときのみ，$[(q_n)]_{\mathrm{C}}$ を定義し，$[(q_n)]_{\mathrm{C}} = \lim_{n\to\infty} q_n$ とする．

■デデキント実数: 有理数の集合 $A \subseteq \mathbb{Q}$ が下方閉とは，任意の $a \in A$ と有理数 $q \leqslant a$ について $q \in A$ となることである．同様に，A が上方閉とは，任意の $a \in A$ と有理数 $q \geqslant a$ について $q \in A$ となることである．**デデキント切断** (*Dedekind cut*) とは，有理数の空でない集合の対 $L, R \subseteq \mathbb{Q}$ で，L は上方閉，R は下方閉，$L \cap R = \varnothing$ かつ $|\mathbb{Q}\backslash(L \cup R)| \leqslant 1$ となることである．この L の開部分として，$L^\circ = \{r \in \mathbb{Q} : (\exists s \in L)\ r < s\}$ と定義する．デデキント切断 (L_0, R_0) と (L_1, R_1) が等しいとは，$L_0^\circ = L_1^\circ$ となることである．

実数の集合 $\mathbb{R}$ の**デデキント表現** (*Dedekind representation*) とは，デデキント切断の枚挙の同値類として実数を表現する関数 $[\cdot]_D :\subseteq (\mathbb{Q}\times\mathbb{Q})^{\mathbb{N}} \to \mathbb{R}$ である．より正確にこの表現を定義するため，$p=(p_n)_{n\in\mathbb{N}}$ と $q=(q_n)_{n\in\mathbb{N}}$ について，p と q がそれぞれ集合 A と B の枚挙であるとき，(p,q) を (A,B) の枚挙と呼び，$\mathrm{Rng}(p,q)=(A,B)$ と書く．このとき，デデキント切断の枚挙 (p,q) に対して，同値類 $[(p,q)]_D$ を以下によって定義する．

$$[(p,q)]_D = \{(r,s) : \mathrm{Rng}(r,s) \text{ は } \mathrm{Rng}(p,q) \text{ と等しいデデキント切断である.}\}$$

(p,q) がデデキント切断 (L,R) の枚挙であるとき，$[(p,q)]_D$ のことを $\sup L$ または $\inf R$ と書く．デテキント切断を用いた表現として，他にも開デデキント表現や決定可能デデキント表現などが知られているが，いずれも計算論を展開する際にはあまり有用でないので，ここでは触れない．

豆知識．ところで，デデキント表現の等しさの定義では，L についての情報しか用いていない．このため，切断の定義は単に L だけ考えればよいのではないか，と思う人もいるかもしれない．そのような表現は**左デデキント表現**などと呼ばれるが，ユークリッド直線 $\mathbb{R}$ の表現としては正しくない．左右からしっかり挟み込まないと，実数の近似精度が評価できないため，位相的に全く異なった概念になってしまう．しかし，$\mathbb{R}$ 上のユークリッド位相の表現として正しくないだけで，$\mathbb{R}$ 上の**上半位相** (*upper topology*) と呼ばれる非 T_1 位相の表現としては正しい．これは $\{0,1\}$ を離散的なオブジェクトとして見るときはシエルピンスキ表現は誤りだが，$\{0,1\}$ を連結空間として見るときはシエルピンスキ表現が正しい，という状況と同様である．

コーシー表現では，実数は有理数のストリームとして表される．デデキント表現では，実数は有理数の対のストリームとして表される．したがって，実数の計算論を展開するための適切な舞台は，ストリーム上の計算論，つまりクリーネの相対第二代数 $\mathbf{K}$ 上の計算論 $\mathsf{Mod}(\mathbf{K})$ である．

このアイデアを用いて，任意精度表現，2 進小数表現，縮小区間表現を，実数のストリーム表現として再定義しよう．

- 2 進小数表現 $[\cdot]_{\mathrm{bin}}$ は，$\{0,1,.\}$ の記号のストリーム α で小数点記号 . が高々 1 つしか含まないものについて，$[\alpha]_{\mathrm{bin}}$ を実数の 2 進表記であると理解する表現である．
- 縮小区間表現 $[\cdot]_I$ は，$\lim_{n\to\infty}(q_n-p_n)=0$ となるような有理数の対のストリーム $(p,q)=(p_n,q_n)_{n\in\mathbb{N}}$ が与えられたとき，$[(p,q)]_I=\lim_{n\to\infty}p_n$ を与える関数である．
- 任意精度表現 $[\cdot]_A$ は，与えられた有理数のストリーム $(\Phi(n))_{n\in\mathbb{N}}$ について，$|x-\Phi(n)|<2^{-n}$ なるような実数 x が存在するとき，そのような x を返す関数である．

ここで，「実数の計算可能性」を定義することと「実数の表現」を与えることの理論的な違い，そして前者よりも後者の方が重要である理屈を説明しよう．まず，「実数の計算可能性」という概念を定義しても，「実関数の計算可能性」はまた個別に定義する必要がある．しかし，「実数の表現」を与えると，そこから「実数の計算可能性」と「実関数の計算可能性」が自動的に定義される．これについて説明しよう．実数の表現を与えた，という事実は，いま $\mathbb{R}$ に $\mathbf{K}$-表現空間としての構造が与えられた，ということである．表現空間上の関数の計算可能性は定義 4.4 で与えた通りである．表現空間の点の計算可能性は以下によって与えられる．

定義 4.43. $(\underset{\sim}{\mathbf{A}}, \mathrm{A})$ を相対部分組合せ代数とする．$\mathbf{A}$-表現空間 X の点 $x \in X$ が**計算可能** (*computable*) であるとは，$x = [a]_X$ となる $a \in \mathbf{A}_\circ$ が存在することを意味する．

例 4.44. クリーネの相対第二代数 $\mathbf{K}$ において，$\mathbf{K}$ はストリーム全体の集合 $\mathbb{N}^{\mathbb{N}}$ であり，$\mathbf{K}_\circ$ は計算可能ストリーム全体の集合，つまり $\mathbb{N}$ 上の計算可能関数全体の集合であった．このとき，$\mathbf{K}$-表現空間 X の点 $x \in X$ が計算可能であるとは，x が計算可能な名を持つことを意味する．

また，実数や実関数の計算可能性だけでなく，たとえば「$\mathbb{R}$ の開部分集合の計算可能性」「$\mathbb{R}$ のコンパクト部分集合の計算可能性」などを含む無数の計算可能性概念も自動的に定義される．このように「表現」を与えるだけで，豊穣な計算可能性理論の世界が自動的に広がっていくのである．

実数の計算論は実数の表現に依存する．したがって，実数のどの表現が同値であり，どの表現が異なる計算論を与えるか，ということは計算可能解析学の初期における問題の 1 つであった．つまり，与えられた実数の計算論が異なる実数の計算論に翻訳可能かどうか，というナンバリングの理論と同様のシチュエーションに辿り着く．ナンバリングの理論のときと同様に，この問題は定義 4.16 で用いた表現の還元可能性の概念を用いて定式化できる．

定理 4.45. 任意精度表現，区間表現，コーシー表現，デデキント表現は同値である．一方，2 進小数表現および素朴コーシー表現は異なる表現であり，以下が成立する．

$$[\cdot]_{\mathrm{bin}} < [\cdot]_A \equiv [\cdot]_I \equiv [\cdot]_C \equiv [\cdot]_D < [\cdot]_{nC}.$$

Proof. $[\cdot]_{\mathrm{bin}} \leqslant [\cdot]_A$ であることは，命題 4.42 の証明において，$[\alpha]_{\mathrm{bin}} = [q]_A$ となることを示したが，$\alpha \mapsto q$ が計算可能であるから，これが $[\cdot]_{\mathrm{bin}} \leqslant [\cdot]_A$ を保証する．$[\cdot]_C \leqslant [\cdot]_{nC}$ は自明である．$[\cdot]_A \equiv [\cdot]_C$ は明らかであろう．$[\cdot]_D \leqslant [\cdot]_I$ について，与えられたデデキント切断の枚挙 $(\ell_n, r_n)_{n\in\mathbb{N}}$ に対して，各 n について $p_n = \max_{s<n} \ell_s$ かつ $q_n = \max_{s<n} r_s$ とすれば，$(p_n, q_n)_{n\in\omega}$ は区間の縮小列であり，自明に $\sup_n \ell_n = \lim_n p_n$ かつ $\inf_n r_n = \lim_n q_n$ であるから，同じ実数を与える．よって，計算可能関数 $(\ell_n, r_n)_{n\in\mathbb{N}} \mapsto (p_n, q_n)_{n\in\omega}$ によって $[\cdot]_D \leqslant [\cdot]_I$ は保証される．$[\cdot]_D \leqslant [\cdot]_I$ も容易に示せる．$[\cdot]_A \leqslant [\cdot]_I$ について，実数の任意精度近似 Φ に対して，区間 $(\Phi(n) - 2^{-n+1}, \Phi(n) + 2^{-n+1})$ を考えればよい．$[\cdot]_I \leqslant [\cdot]_A$ について，区間縮小列 $(p_n, q_n)_{n\in\mathbb{N}}$ が与えられたとき，各 n に対して $q_s - p_s < 2^{-n}$ なる $s \in \mathbb{N}$ を探し，$\Phi(n)$ をその区間 (p_s, q_s) の中点，つまり $\Phi(n) = (q_s - p_s)/2$ と定義すればよい．

次に $[\cdot]_{nC} \not\leqslant [\cdot]_A$ を示そう．H_s を停止問題の時刻 s 近似，つまり $H_s = \{e < s : [\![e]\!](e)[s] \downarrow\}$ とする．このとき $q_s = \sum_{e\in H_s} 2^{-e}$ と定義すると，$(q_s)_{s\in\mathbb{N}}$ は計算可能なコーシー列である．特に，その極限 $x = \lim_{s\to\infty} q_s$ は，素朴コーシー表現において計算可能である．しかし，x の 2 進表記を見ると，明らかに停止問題 Halt の情報を記している．停止問題の計算不可能性より，これは x が $(\mathbb{R}, [\cdot]_{\mathrm{bin}})$ で計算不可能であることを意味する．一方，命題 4.42 より，x は任意精度表現でも計算

不可能である．よって，$[\cdot]_{nC} \not\equiv [\cdot]_A$ を得る．既に $[\cdot]_A \leqslant [\cdot]_{nC}$ は示してあるから，$[\cdot]_{nC} \not\leqslant [\cdot]_A$ である．

最後に，$[\cdot]_A \not\leqslant [\cdot]_{\mathrm{bin}}$ を示す．$\mathbb{R}_{\mathrm{bin}}$ を表現空間 $(\mathbb{R}, [\cdot]_{\mathrm{bin}})$ とすると，$f(x) = 3x$ で定義された関数 $f : \mathbb{R}_{\mathrm{bin}} \to \mathbb{R}_{\mathrm{bin}}$ が計算不可能であることを示す．他の表現の場合は，$x \mapsto 3x$ は容易に計算可能であることが分かるので，これが 2 進小数表現と他の表現の違いを導く．もし $f(x) = 3x$ が 2 進小数表現の元で計算可能だったとする．$1/3$ を 2 進小数展開すると $0.01010101\ldots$ という循環小数になることに注意する．f を実現するチューリング機械 M が，入力ストリーム $0.01010101\ldots$ を読み込んでいるとしよう．このとき，M は必ずある時点でストリームを出力し始めなければならない．入力ストリームの n 桁目を読み込んだ段階で，M がある文字 k を出力したとしよう．M は $x \mapsto 3x$ を計算しているはずなので，$k = 1$ であるか $k = 0$ であり，その後に小数点とそれ以下の値が続くはずである．もし $k = 1$ だったとしたら，入力ストリームの $n + 1$ 桁以降が何であろうと，M は 1 以上の実数を記述する．しかし，$n + 1$ 桁目以降ずっと 0 を返すストリーム α を考えると，$[\alpha]_{\mathrm{bin}} < 1/3$ であるから，$f([\alpha]_{\mathrm{bin}}) < 1 \leqslant [[\![M]\!](\alpha)]_{\mathrm{bin}}$ となり，M は f を正しく計算できていない．もし $k = 0$ だったとしたら，入力ストリームの $n + 1$ 桁以降が何であろうと，M は 1 以下の実数を記述する．しかし，$n + 1$ 桁目以降ずっと 1 を返すストリーム α を考えると，$[\alpha]_{\mathrm{bin}} > 1/3$ であるから，$f([\alpha]_{\mathrm{bin}}) > 1 \geqslant [[\![M]\!](\alpha)]_{\mathrm{bin}}$ となり，M は f を正しく計算できていない．よって，$x \mapsto 3x$ が 2 進小数表現の下では計算不可能であることが示された．以上より，$[\cdot]_{\mathrm{bin}}$ が他の表現と同値でないことが示されるが，既に $\leqslant [\cdot]_{\mathrm{bin}} \leqslant [\cdot]_A$ は示してあるから，$[\cdot]_A \not\leqslant [\cdot]_{\mathrm{bin}}$ を得る． □

この定理の興味深いところは，悪い表現，というものにも複数の方向性があり，表現の還元可能性概念がそれを明示してくれる点にある．つまり 2 進小数表現は，要求が厳しすぎる表現であり，素朴コーシー表現は，要求が甘すぎる表現であると言える．

豆知識．本節で述べた実数の計算モデルは，**厳格実数計算** (*exact real computation*), **誤差なし実数計算** (*error-free real computation*), あるいは**任意精度計算**などとも呼ばれる．最近は，幾つかのプログラミング言語で厳格実数計算用のライブラリが少しずつ充実しつつあるようだ．

豆知識．実数の計算論には，かつては他にも BSS 機械 (Blum-Shub-Smale machine) という計算モデルなどがあった．厳格実数計算モデルが「位相空間論的」実数の計算論と考えられるのに対し，BSS モデルは「モデル理論的」実数の計算論と思える．しかし，計算論として取り扱うとなると，BSS モデルは若干微妙なところがある．たとえば，厳格実数計算とは異なり，BSS 計算は現実のコンピュータで実装できない，実数の位相構造と相性が悪い，数学的にも美しさが足りない，などが代表的な問題点である．このため，BSS 機械は，現在ではあまり良い計算モデルとは考えられなくなってしまった．

4.4 ベール表現空間の理論 ⋆

任意の部分組合せ代数 $\mathbf{A}$ に対して，$\mathbf{A}$-表現空間の概念を考えることができるが，近年において，最も主流な研究対象は $\mathbf{A}$ がクリーネの（相対）第二代数 $\mathbf{K}$ の場合である．クリーネの第二代数 $\mathbf{K}$ の表現空間論 $\mathsf{Mod}(\mathbf{K})$ における重要な問題は，どのような空間が良い意味で表現できるか，とい

う問題である．たとえば，可分距離空間の計算可能性理論は極めて長い歴史を持つ．

可分距離空間であれば，第 4.3 節で学んだ実数の計算可能性理論を適用可能であることは容易に分かるだろう．具体的には，可分距離空間 X の距離関数 d と可算稠密部分集合 $Q \subseteq X$ を用いて，コーシー表現を用いるなどがひとつの方法である．コーシー表現は，多くの場合に上手く行くが，しばしばより良い表現が必要になる場合もある．したがって，次の記述集合論的概念を用いた表現を考えよう．

定義 4.46. 空間 X 上の**スリン・スキーム** (*Suslin scheme*) は $\mathbb{N}^*$ で添字付けられた集合族 $\mathcal{S} = (S_\sigma)_{\sigma\in\mathbb{N}^*}$ を表す．距離空間 X 上のスリン・スキーム $\mathcal{S}$ が**直径消失** (*vanishing diameter*) とは，任意の $x \in \mathbb{N}^{\mathbb{N}}$ について $\lim_{n\to\infty} \operatorname{diam}(S_{x\restriction n}) = 0$ となることである．また，$\operatorname{diam}(S_{x\restriction n}) \leqslant 2^{-n}$ であれば，**急消失**であると呼ぶ．距離空間 X の**スリン・スキーム表現**は，以下を満たす急消失スリン・スキーム $\mathcal{S} = (S_\sigma)_{\sigma\in\mathbb{N}^*}$ である．

1. $S_\varepsilon = X$ かつ任意の $\sigma \in \mathbb{N}^*$ について S_σ は空でない開集合である．
2. 任意の $\sigma \in \mathbb{N}^*$ について，$\operatorname{cl}(S_{\sigma^\frown n}) \subseteq S_\sigma = \bigcup_n S_{\sigma^\frown n}$ である．

補題 4.47. 任意の可分距離空間はスリン・スキーム表現を持つ．

Proof. a_e を e 番目の有理点，q_n を n 番目の有理数とする．$I_\sigma = \{\langle e,k\rangle : \overline{B}(a_e;q_k) \subseteq S_\sigma \text{ and } q_k \leqslant 2^{-n-1}\}$ とし，$(e(n),k(n))$ を I_σ の n 番目の要素とする．このとき，$S_{\sigma^\frown n} = B(a_{e(n)};q_{k(n)})$ と定義する．まず $\operatorname{cl}(S_{\sigma^\frown n}) \subseteq \overline{B}(a_e;q_k) \subseteq S_\sigma$ である．任意の $x \in S_\sigma$ に対して，S_σ は開集合であるから，X の正則性より，ある $\varepsilon > 0$ が存在して，$\overline{B}(x;\varepsilon) \subseteq S_\sigma$ となる．ある $e,k \in \mathbb{N}$ について，$|x - a_e| < q_k \leqslant \varepsilon/2$ となるので，$B(a_e;q_k) \subseteq B(x;\varepsilon)$ であるから，$(e_n,k_n) = (e,k)$ なる n を取れば，$x \in S_{\sigma^\frown n}$ を得る．よって，$S_\sigma = \bigcup_n S_{\sigma^\frown n}$ である． □

距離空間 X のスリン・スキーム表現 $\mathcal{S}$ は，自明に $\mathbf{K}$-表現を誘導する．具体的には，$x \in \mathbb{N}^{\mathbb{N}}$ に対して，$\bigcap_{n\in\mathbb{N}} S_{x\restriction n}$ が空でないなら，$\nu^{\mathcal{S}}(x)$ をその唯一の元として定義する．スリン・スキームの誘導する $\mathbf{K}$-表現 $\nu^{\mathcal{S}} :\subseteq \mathbb{N}^{\mathbb{N}} \to X$ の優れた点は，まず連続かつ**開写像** (*open map*) であるという点である．続いて，もし X が完備可分距離空間ならば，$\nu^{\mathcal{S}}$ が全域 $\mathbf{K}$-表現である．この 3 つの全域開連続性は位相的性質であるから，これは任意のポーランド空間の全域連続開 $\mathbf{K}$-表現を与えるということである．ここで，位相空間が**ポーランド空間** (*Polish space*) であるとは，完備可分距離化可能であることを意味する．

マシュー・デ・ブレクト (Matthew de Brecht) は，距離化不可能空間でも同様の表現を持ち得ることに気づいた．デ・ブレクトは**擬ポーランド空間** (*quasi-Polish space*) の概念を導入した．擬ポーランド空間とは，スミス完備擬距離化可能 (Smyth-completely quasi-metrizable) な第二可算空間である．任意のポーランド空間は擬ポーランド空間であるが，ポーランド空間でない自然な擬ポーランド空間が無数にあることが知られている．擬ポーランド空間は一般には距離化可能でないどころか T_1-分離公理さえ満たさない．デ・ブレクトは，擬ポーランド空間の記述集合論を創始し，

以下の特徴付けを与えた.

定理 4.48 (デ・ブレクトの定理). 位相空間が擬ポーランド空間であることと全域開連続 **K**-表現を持つことは同値である.

さて，全域開連続 **K**-表現が良い表現であることは疑いようがないが，どのような表現であれば位相空間の良い表現であるといえるだろうか．我々は良い「**位相計算構造**」を求めている．位相群であれば基本的な群演算が連続であるように，位相計算構造において基本的な計算は連続であるべきだろう．つまり，圏 $\mathsf{Mod}(\mathbf{K})$ における射と位相空間における何らかの連続性に対応があってしかるべきである.

これに対する当初の解答は，第二可算 T_0 空間に焦点を絞って考えられていた．第二可算 T_0 空間の近傍フィルター表現を考えると，十分に良い性質を持つ．したがって，任意の第二可算 T_0 空間に関する計算可能性理論が容易に展開できることは分かっていた．しかし，第二可算空間への制限は計算理論に対する強すぎる縛りだということが次第に判明する.

これは高階計算論の文脈でより明確になる．1950 年代にクリーネは高階関数空間の計算論を展開するために，**アソシエイト** (*associate*) の概念を導入した．クリーネは**主アソシエイト** (*principal associate*) というものも考えたが，これは高階関数空間においても原始的概念を開集合のように扱おうとするものであった．しかし，これは上手く行かないことが分かり，主アソシエイトの概念はすぐに放棄された．現代的な観点から，クリーネのアプローチを述べると，主アソシエイトは高階関数空間に可算開基を無理に導入し，第二可算空間のように扱おうとする試みであった．一方，アソシエイトの概念は，原始的概念が開集合であると考えることを諦め，ジェネラル・トポロジーの言葉を借りれば，ある種の「可算ネットワーク」を導入する考え方である.

このような第二可算公理を満たし得ない空間に対する計算可能性理論の展開のために，マティアス・シュレーダー (Matthias Schröder) は認容表現の概念を導入した.

定義 4.49. 位相空間 X の**認容表現** (*admissible representation*) とは，連続 **K**-表現 $\nu :\subseteq \mathbb{N}^{\mathbb{N}} \to X$ であり，次の普遍性を持つものである．任意の連続関数 $\delta :\subseteq \mathbb{N}^{\mathbb{N}} \to X$ に対して，ある $\tau :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ が存在して，$\delta = \nu \circ \tau$ を満たす.

スースリン・スキームの誘導する表現は認容表現であることを示すことができる．したがって，擬ポーランド空間であることと全域開連続認容表現を持つことは同値である．事実，デ・ブレクトは全域認容表現を持つだけで擬ポーランド性が導かれることを示した.

それでは，どのような空間が全域とは限らない認容表現を持ち得るだろうか．ところでジェネラル・トポロジーにおいて，1959 年にアレクサンダー・アルハンゲルスキ (Alexander Arhangel'skii) は，**開基** (*open basis*) の概念を一般化する**ネットワーク** (*network*) の概念を提唱した．可算ネットワークを持つ正則 T_1 空間は，**コズミック空間** (*cosmic space*) の名の下で深く研究されている．コズミック空間という名称は言葉遊びであって，宇宙空間とはあまり関係ない．名前の由来は，コズ

表 1　様々な位相空間の関係

コズミック空間		
⇑		
$\aleph_0$ 空間	$\Longrightarrow$	認容表現空間
⇑		⇑
可分距離化可能空間	$\Longrightarrow$	第二可算 T_0 空間
⇑		⇑
ポーランド空間	$\Longrightarrow$	擬ポーランド空間

ミック空間は可分距離空間の連続像となるからである．

1960 年代以降になると，**k-ネットワーク** (k-*network*) や**収束列ネットワーク** (*cs-network*) などの変種が研究されるようになった．可算 k-ネットワークを持つ正則 T_1 空間は，**$\aleph_0$-空間** ($\aleph_0$-*space*) と呼ばれる．正則 T_1 空間が可算 k-ネットワークを持つことと可算収束列ネットワークを持つことは同値であることは知られている．

シュレーダーは，認容表現空間の基本定理とも呼べる以下の結果を示した．

定理 4.50 (シュレーダーの定理)**.** T_0 空間 X について，以下は同値である．

1. X は認容表現を持つ．
2. X は可算収束列ネットワークを持つ．
3. X は可分距離空間の商空間となる．

したがって，現代的な計算可能性理論においては，開基よりもある種のネットワークを原始的な概念として取り扱う方が自然である．シュレーダーの定理の重要な点は，T_1-性や正則性などの強い分離公理を仮定していない点である．実際，計算可能性理論に現れる多くの空間は，そのような強い分離公理を満たさない．

豆知識． $\mathbf{K}$-表現空間の理論は，**型 2 実効理論** (*type two theory of effectivity*) と呼ばれることもある．しかし，もちろん $\mathsf{Mod}(\mathbf{K})$ はデカルト閉圏をなし，高階関数空間は型 2 実効理論における中心的な研究対象の 1 つである．このため，この理論を「型 2」という名称で呼ぶのは若干，誤解を招く可能性があるように思う．似たような例として，二階算術も同様の誤解を受けることがあるようである．

5　極限計算可能性

5.1　極限計算と極限補題

我々は,「計算」というものを時間経過毎に変化する計算状況の極限と考えることができる．計算モデルとして再びチューリング機械を考えよう．定義 1.3 の計算状況の記法を思い返すと，チュー

リング機械 M に $\sigma \in \Sigma^*$ を入力した場合の計算過程は，

$$M(\sigma)[0] \longrightarrow_M M(\sigma)[1] \longrightarrow_M M(\sigma)[2] \longrightarrow_M M(\sigma)[3] \longrightarrow_M \dots$$

となっていく．すると，チューリング機械 M に σ を入力した計算が停止するということは，計算状況の列 $(M(\sigma)[s])_{s\in\omega}$ の値が何らかの状況に**収束** (*converge*) するということである．言い換えれば，極限 $\lim_{s\to\infty} M(\sigma)[s]$ が存在する，ということである．逆に，M に σ を入力した計算が停止しないということは，列 $(M(\sigma)[s])_{s\in\omega}$ が**発散** (*diverge*) するということであり，つまり，極限 $\lim_{s\to\infty} M(\sigma)[s]$ は存在しない，ということである．このように計算概念と極限概念には深い関わりがある．

計算状況が受理状態に辿り着いたことを 1 で表し，まだ受理状態に辿り着いていないことを 0 で表すことによって単純化しよう．すると，停止する計算と停止しない計算はそれぞれ以下のように表される．

$$\begin{aligned}
\text{停止する計算:} &\quad 0 \longrightarrow 0 \longrightarrow 0 \longrightarrow \dots \longrightarrow 0 \longrightarrow 1 \longrightarrow 1 \longrightarrow 1 \longrightarrow \dots \\
\text{停止しない計算:} &\quad 0 \longrightarrow 0 \longrightarrow 0 \longrightarrow \dots \longrightarrow 0 \longrightarrow 0 \longrightarrow 0 \longrightarrow 0 \longrightarrow \dots
\end{aligned}$$

このように単純化すると，停止しない計算でも 0 に収束してしまっているが，物事を先程より粗く見たいので，ここではそれで構わない．

さて，半計算可能性の定義 1.32 を思い出そう．集合 A が半計算可能であるということは，あるチューリング機械 M が存在して，$n \in A$ であることと $[\![M]\!](n)$ が停止することが同値である，つまり，上の計算列が途中で一度だけ 0 から 1 に切り替わる，ということである．つまり，半計算可能性は「初期値 0 心変わり 1 の極限計算可能性」として特徴づけられる．

命題 5.1. 集合 $A \subseteq \mathbb{N}$ が半計算可能であることと，次の 3 条件を満たす計算可能関数 $\varphi : \mathbb{N} \times \mathbb{N} \to \{0, 1\}$ が存在することは同値である．

$$\begin{aligned}
\text{極限計算可能性:} &\quad k \in A \iff \lim_{s\to\infty} \varphi(k, s) = 1 \\
\text{初期値 0:} &\quad \varphi(k, 0) = 0 \\
\text{変心 1:} &\quad |\{s \in \mathbb{N} : \varphi(k, s) \neq \varphi(k, s+1)\}| \leqslant 1
\end{aligned}$$

Proof. 上で述べたことを厳密に書き下すだけである．$A \subseteq \mathbb{N}$ が半計算可能であるならば，あるチューリング機械 M が存在して，$n \in A$ であることと $M(n)\downarrow$ であることが同値である．このとき，

$$\varphi(n, s) = \begin{cases} 1 & \text{if } M(n)[s]\downarrow \\ 0 & \text{if } M(n)[s]\uparrow \end{cases}$$

とする．$M(n)$ の計算を s ステップだけシミュレートすれば $\varphi(n, s)$ の値を求められるので，φ は計算可能である．このとき，明らかに $n \in A$ であることと $\lim_n \varphi(n, s) = 1$ であることは同値である．また，$\varphi(n, s) = 0$ であり，変心 1 回であることも明らかである．

逆に，主張の 3 条件を満たす φ が与えられているとする．このとき，

$$M(k)\downarrow \iff (\exists s\in\mathbb{N})\ \varphi(k,s)=1$$

となるチューリング機械 M を作ることができる．たとえば，入力 k に対して，$\varphi(k,0),\varphi(k,1),\ldots$ の値を計算していき，もし $\varphi(k,s)=1$ なる $s\in\mathbb{N}$ なる s が見つかったら計算を停止すればよい．このとき，

$$k\in A \iff \lim_{s\to\infty}\varphi(k,s)=1 \iff (\exists s\in\mathbb{N})\ \varphi(k,s)=1 \iff M(k)\downarrow$$

であるから，A は半計算可能である． □

命題 5.1 の 3 条件は，上から順に，**極限計算可能性** (*limit computability*), 初期値 0, **変心** 1 (*at most one mind-change*) を表す．すると，より多くの心変わりを行う計算モデルを考えたらどうなるだろうか．このような計算モデルは，確定的に計算できなくとも，近似的に計算できれば十分，という状況の下で便利である．

定義 5.2. 関数 $g:\mathbb{N}\to\mathbb{N}$ が**極限計算可能** (*limit computable*) とは，次を満たす計算可能関数 $\tilde{g}:\mathbb{N}\times\mathbb{N}\to\mathbb{N}$ が存在することである．

$$g(k)=\lim_{s\to\infty}\tilde{g}(k,s).$$

例 5.3 (極限同定の理論)**.** 1967 年，ゴールド (E. Mark Gold) は言語を学習する人工知能の数学的モデル化のために，極限計算可能性の概念を用いた．ゴールドの理論は**極限同定** (*identification in the limit*) の理論と呼ばれ，その後，機械学習の数学的枠組を研究する計算論的学習理論の一分野として発展した．

命題 2.3 を思い返すと，停止問題に多対一還元されることと，半計算可能であることは同値であった．命題 5.1 と組み合わせると，

$$A\leqslant_m \mathsf{Halt} \iff A \text{ は初期値 0 変心 1 極限計算可能である．}$$

定理 5.4 (極限補題)**.** 任意の関数 $g:\mathbb{N}\to\mathbb{N}$ について，次が成立する．

$$g\leqslant_T \mathsf{Halt} \iff g \text{ は極限計算可能である．}$$

Proof. $g\leqslant_T \mathsf{Halt}$ を仮定すると，ある部分計算可能関数 $\Phi:\subseteq \mathbf{2}^{\mathbb{N}}\to\mathbb{N}^{\mathbb{N}}$ が存在して $\Phi(\mathsf{Halt})=g$ となる．停止問題 Halt は計算可能なので，命題 5.1 より，近似 h が存在する．このとき，$\mathsf{Halt}_s=\{n<s : h(n,s)=1\}$ と定義し，これは長さ s の文字列と同一視する．定義 3.2 の直後の注意よ

り，Φ を生成する全域計算可能単調関数 $\varphi : \mathbf{2}^* \to \mathbb{N}^*$ が存在する．このとき，

$$\tilde{g}(n,s) = \begin{cases} \varphi(\mathsf{Halt}_s)(n) & \text{if } \varphi(\mathsf{Halt}_s)(n)\downarrow \\ 0 & \text{otherwise} \end{cases}$$

と定義すれば，$\tilde{g}$ は計算可能である．いま，任意の $n \in \mathbb{N}$ について，$\Phi(\mathsf{Halt})(n) = g(n)$ であるから，ある $t \in \mathbb{N}$ が存在して $\varphi(\mathsf{Halt} \upharpoonright t)(n) = g(n)$ となる．ここで $A \upharpoonright t = \{n < t : n \in A\}$ を表す．明らかに $\mathsf{Halt} \upharpoonright t = \lim_s(\mathsf{Halt}_s \upharpoonright t)$ であるから，十分大きな任意の s について，

$$\tilde{g}(n,s) = \varphi(\mathsf{Halt}_s \upharpoonright t)(n) = \varphi(\mathsf{Halt} \upharpoonright t)(n) = g(n)$$

が成立するから，$\lim_s \tilde{g}(n,s) = g(n)$ である．

逆に，$\lim_s \tilde{g}(n,s) = g(n)$ を仮定する．まず，次のような集合 $S \subseteq \mathbb{N}$ は半計算可能である．

$$\langle n,k\rangle \in S \iff |\{s \in \mathbb{N} : g(n,s+1) \neq g(n,s)\}| > k.$$

なぜなら，入力 $\langle n,k\rangle$ が与えられたとき，$g(n,\cdot)$ が k 回変心するまで待機し，そうなったら停止するチューリング機械を考えればよい．命題 2.3 より，停止問題は Σ_1-完全なので，$S \leqslant_m \mathsf{Halt}$ を保証する計算可能関数 $f : \mathbb{N} \to \mathbb{N}$ が存在する．つまり，$|\{s \in \mathbb{N} : g(n,s+1) \neq g(n,s)\}| > k$ であることと $f(n,k) \in \mathsf{Halt}$ であることが同値である．神託チューリング機械 M として，次のようなものを考える．入力 n に対して，$f(n,k) \notin \mathsf{Oracle}$ となる最小の k を探索し，そのような k を得られたら，$g(n,s)$ が k 回変心するまで待ち，その結果を返す．このとき，$[\![M]\!](\mathsf{Halt}; n) = g(n)$ を得る． □

豆知識． 極限補題は，1959 年にシェーンフィールドによって，1965 年にパトナムによって独立に証明された．シェーンフィールドが示した極限補題はもう少し強い形で，収束率に関する有用な洞察をさらに含んでおり，この強い形の方を極限補題と呼ぶことも多い．

5.2 パトナムの試行錯誤の階層

定義 5.5. 関数 $A : \mathbb{N} \to \mathbb{N}$ が **n-変心計算可能** (*computable with n mind changes*) とは，次の 2 条件を満たす計算可能関数 $\varphi : \mathbb{N} \times \mathbb{N} \to \{0,1\}$ が存在することである．

$$\begin{aligned} \text{極限計算可能性:} &\quad A(k) = \lim_{s\to\infty} \varphi(k,s) \\ \text{変心 } n\text{:} &\quad |\{s \in \mathbb{N} : \varphi(k,s) \neq \varphi(k,s+1)\}| \leqslant n \end{aligned}$$

A が集合であり，φ が初期値条件 $\varphi(k,0) = 0$ を満たす場合，A を **n-半計算可能**と呼び，初期値条件 $\varphi(k,0) = 1$ を満たす場合，**n-余半計算可能**と呼ぶ．

豆知識． n-変心計算可能集合は，パトナムによって **n-試行錯誤述語** (*n-trial-and-error predicate*) と呼ばれていたものである．n-半計算可能性および n-余半計算可能性は，それぞれ n-c.e. および co-n-c.e. という名で呼ばれることが多い．

また，$A \subseteq \mathbb{N}$ が 2-半計算可能であることと，ある半計算可能集合 $B, C \subseteq \mathbb{N}$ が存在して $A = B \backslash C$ となることは同値であるので，2-半計算可能性のことを d-c.e. と呼ぶことも多い．ここで，d は**差** (*difference*) を意味する．

定理 5.6. 集合 $A \subseteq \mathbb{N}$ について，次の 2 条件は同値である．

1. A は n-変心計算可能である．
2. A は $(n+1)$-半計算可能かつ $(n+1)$-余半計算可能である．

この証明のために，$(n+1)$-余半計算可能の**分離性** (*separation property*) と呼ばれる性質を証明する．

補題 5.7. $A, B \subseteq \mathbb{N}$ を $A \cap B = \varnothing$ であるような $(n+1)$-余半計算可能集合とする．このとき，次のような n-変心計算可能集合 $C \subseteq \mathbb{N}$ が存在する．

$$A \subseteq C, \text{ and } B \cap C = \varnothing.$$

Proof. $A, B \subseteq \mathbb{N}$ を $(n+1)$-余半計算可能集合で，$A \cap B = \varnothing$ なるものとする．$p, q : \mathbb{N}^2 \to \{0,1\}$ をそれぞれ A と B の初期値 1, 変心 $n+1$ 近似とする．$k \in \mathbb{N}$ を固定する．$A \cap B = \varnothing$ なので，$p(k, s_k) = 0$ または $q(k, s_k) = 0$ なる $s_k \in \mathbb{N}$ が存在する．そのような最小の $s_k \in \mathbb{N}$ を取る．このとき，以下のように $r : \mathbb{N}^2 \to \{0,1\}$ を定義する．

$$r(k,t) = \begin{cases} p(k, s_k + t) & \text{if } p(k, s_k) = 0 \\ 1 - q(k, s_k + t) & \text{otherwise.} \end{cases}$$

前者の場合，$p(k, s_k) \neq p(k, 0)$ であり，後者の場合，$q(k, s_k) \neq q(k, 0)$ であるから，既に 1 回以上変心している．よって，s_k 以降では，残り n 回以下しか変心できない．つまり，r は高々 n 回しか変心しない．$C(k) = \lim_t r(k,t)$ によって定義すると，C は n-変心計算可能である．このとき，

$$\begin{aligned} p(k, s_k) = 0 &\implies C(k) = \lim_{t\to\infty} r(k,t) = \lim_{t\to\infty} p(k, s_k + t) = A(k), \\ p(k, s_k) \neq 0 &\implies C(k) = \lim_{t\to\infty} r(k,t) = \lim_{t\to\infty} (1 - q(k, s_k + t)) = 1 - B(k). \end{aligned}$$

よって，$A \subseteq C$ かつ $C \cap B = \varnothing$ が成り立つ．つまり，A と B は n-変心計算可能集合 C によって分離される． □

*Proof (*定理 5.6*).* (1)⇒(2): φ を A の変心 n 近似とする．このとき，各 $i \in \{0,1\}$ について，$\psi_i(n, 0) = i$ かつ $\psi_i(n, s+1) = \varphi(n, s)$ によって定義すれば，ψ_i は A の初期値 i 変心 $(n+1)$ 近似である．よって，A は $(n+1)$-半計算可能かつ $(n+1)$-余半計算可能である．

(2)⇒(1): A は $(n+1)$-半計算可能かつ $(n+1)$-余半計算可能であるということは，A と $\mathbb{N} \backslash A$ が共に $(n+1)$-余半計算可能であるということである．よって，補題 5.7 より，n-変心計算可能集合

$C \subseteq \mathbb{N}$ で $A \subseteq C$ かつ $(\mathbb{N} \backslash A) \cap C = \varnothing$ なるものが存在するが，これは $C = A$ を導く．よって，A は n-変心計算可能である． □

1980 年代になると，エルショフ (Yuri Ershov) は，パトナムの階層を順序数変心の極限計算可能性に拡張した．この階層は現在，**エルショフ階層** (*Ershov hierarchy*) と呼ばれ，計算可能性理論の基本的概念となっている．変心の階層を順序数に拡張する必要性について疑問を抱く人もいるかもしれないが，実はこの拡張は理論的には極めて重要である．というのも，変心の階層を順序数に拡張することによって，一般位相空間論における**ハウスドルフの差の階層** (*Hausdorff difference hierarchy*) と結び付くことができ，計算可能性理論の位相空間的解釈あるいはその逆をより完全なものに近づけるからである．しかし，エルショフ階層を正確に定義するためには，計算可能順序数の理論が必要であるため，ここではエルショフ階層の第 ω 階のみ議論する．

定義 5.8. 関数 $A : \mathbb{N} \to \mathbb{N}$ が **ω-変心計算可能** (*computable with ω mind changes*) とは，次の 2 条件を満たす計算可能関数 $\varphi : \mathbb{N} \times \mathbb{N} \to \{0, 1\}$ と $h : \mathbb{N} \to \mathbb{N}$ が存在することである．

$$\begin{array}{rc} \text{極限計算可能性:} & A(k) = \lim_{s \to \infty} \varphi(k, s) \\ \text{計算可能変心:} & |\{s \in \mathbb{N} : \varphi(k, s) \neq \varphi(k, s+1)\}| \leqslant h(n) \end{array}$$

ω-変心計算可能集合 $A \subseteq \mathbb{N}$ は通常 ω-c.e. と呼ばれる．

定義 5.9. 部分単調関数 $\varphi : \mathbb{A}^* \to \mathbb{A}^*$ の**連続率** (*modulus of continuity*) とは，任意の $\sigma \in \mathrm{dom}(\varphi)$ について $|\sigma| \leqslant h(|\varphi(\sigma)|)$ を満たす関数 $m : \mathbb{N} \to \mathbb{N}$ である．

関数 $f :\subseteq \mathbf{2}^{\mathbb{N}} \to \mathbf{2}^{\mathbb{N}}$ が**部分計算可能一様連続**であるとは，計算可能連続率を持つ部分計算可能単調関数 φ について $f = \widehat{\varphi}$ となることである．ここで定義 3.1 のように，$\widehat{\varphi}$ は単調関数 φ が生成する連続関数を意味する．同様に，関数 $f : \mathbf{2}^{\mathbb{N}} \to \mathbf{2}^{\mathbb{N}}$ が**全域計算可能一様連続**であるとは，計算可能連続率を持つ全域単調関数 φ について $f = \widehat{\varphi}$ となることである．

定義 5.10. 集合 $A, B \subseteq \mathbb{N}$ について，$A \leqslant_{wtt} B$ であるとは，ある部分計算可能一様連続関数 $f :\subseteq \mathbf{2}^{\mathbb{N}} \to \mathbf{2}^{\mathbb{N}}$ が存在して，$f(B) = A$ となることである．同様に，$A \leqslant_{tt} B$ であるとは，ある全域計算可能一様連続関数 $f : \mathbf{2}^{\mathbb{N}} \to \mathbf{2}^{\mathbb{N}}$ が存在して，$f(B) = A$ となることである．

豆知識． 歴史的には，tt-還元可能性 $\leqslant_{tt}$ は**真理値表** (*truth-table*) を用いて導入されていたので，**真理値表還元** (*truth-table reducible*) と呼ばれていた．同様に，wtt は**弱真理値表** (*weak truth-table*) の省略である．しかし，現代的には，この概念を真理値表によって取り扱う文献はほとんど見ない．実際，ここでの定義のように一様連続性の計算可能バージョンとして取り扱う方が見通しも良い．

感覚的には，wtt-還元は，神託へのアクセス範囲が計算可能に抑えられているようなチューリング還元である．さらに，tt-還元は，それに加えて，神託チューリング機械の計算時間が事前に計算可能に抑えられてい

るようなチューリング還元である．特に，以下が成立する．

$$A \leqslant_m B \implies A \leqslant_{tt} B \implies A \leqslant_{wtt} B \implies A \leqslant_T B.$$

これらの矢印の逆のいずれも，一般には成立しない．

5.3 実数の極限計算可能性

実数に関する極限計算可能性概念を導入しよう．

定義 5.11. 実数 $x \in \mathbb{R}$ が**極限計算可能** (*limit computable*) であるとは，ある実数の計算可能列 $(x_n)_{n\in\omega}$ が存在して，$x = \lim_n x_n$ となることである．

もし，この $(x_n)_{n\in\omega}$ が上昇列として取れるならば，x は**下半計算可能**または**左計算可枚挙** (*left-c.e.*) と呼ばれる．同様に，$(x_n)_{n\in\omega}$ が下降列であれば，x は**上半計算可能**または**右計算可枚挙** (*right-c.e.*) と呼ばれる．

集合 $A \subseteq \mathbb{N}$ を 2 進無限列と同一視すると，

$$0.A(0)A(1)A(2)A(3)\ldots A(n)A(n+1)\ldots$$

は，ある実数 $x \in [0,1]$ の 2 進展開を表しているものと考えられる．この実数を $0.A$ と表すこととしよう．このとき，実数 $x \in \mathbb{R}$ が 2 **進半計算可能** (*binary c.e.*) とは，ある半計算可能集合 $A \subseteq \mathbb{N}$ について，$x = 0.A$ となっていることを意味する．同様の方法で，2 進極限計算可能性，2 進 n-変心計算可能性などを定義する．

演習問題 5.12. 任意の 2 進半計算可能実数は下半計算可能であることを示せ．

演習問題 5.13. 実数が極限計算可能であることと 2 進極限計算可能であることは同値であることを示せ．

命題 5.14. 計算不可能な下半計算可能実数が存在する．

Proof. $0.\mathsf{Halt}$ は計算不可能な 2 進半計算可能実数である．演習問題 5.12 より，$0.\mathsf{Halt}$ は下半計算不可能である． □

豆知識. 下半計算可能性の定義より，命題 5.14 から，計算不可能実数に収束する計算可能上昇列 $(x_n)_{n\in\omega}$ が存在することがわかる．そのような列 $(x_n)_{n\in\omega}$ は**スペッカー列** (*Specker sequence*) と呼ばれることがある．スペッカー列の存在は今となっては計算可能性理論における最も自明な結果のひとつとなった．しかしこれをスペッカー (Ernst Specker) が証明したのは 1949 年のことであり，構成的解析学（直観主義論理上の解析学）における証明不可能性の研究への応用があってこそのものである．スペッカーの結果は，計算可能性理論単独のトピックとしては特筆すべきところはないが，構成的解析学という文脈においては，計算可能性理論を応用した証明不可能性証明の最初期の結果としての価値は大きい．また，これは逆数学と呼ばれる分野の揺籃期における歴史的な最初のステップとして考えることもできるであろう．

命題 5.15. 任意の下半計算可能実数は 2 進 ω-変心計算可能である．

Proof. $0.A$ を下半計算可能実数とすると，実数の計算可能上昇列 $0.A_s$ が存在する．このとき，任意 $s \leqslant t$ について，$0.A_s \leqslant 0.A_{s+1}$ であるから，もし $0 = A_t(n) < A_s(n) = 1$ ならば，ある $m < n$ について $1 = A_t(m) > A_s(m) = 0$ でなければならない．よって，$(A_s(n))_{s\in\mathbb{N}}$ は高々 2^n 回しか変心できない．$0.A = \lim_s 0.A_s$ であるから，$0.A$ は 2 進 ω-変心計算可能である． □

例 5.16. 任意の $n \in \mathbb{N}$ について，2 進 n-変心計算可能でない下半計算可能実数が存在することも示せる．これはたとえばチャイティンの Ω などが該当する．

実数 $x \in \mathbb{R}$ が**弱計算可能** (*weakly computable*) とは，次のような計算可能な有理数列 $(x_n)_{n\in\omega}$ が存在することである．

$$x = \lim_{n\to\infty} x_n, \text{ and } \sum_{n=0}^{\infty} |x_{n+1} - x_n| < \infty$$

定理 5.17. 実数 $x \in \mathbb{R}$ について，次の 2 条件は同値である．

1. x は弱計算可能である．
2. $x = y - z$ となる下半計算可能実数 $y, z \in \mathbb{R}$ が存在する．

Proof. (1)⇒(2): $(x_n)_{n\in\omega}$ を x に収束する有理数の計算可能列で，$\sum_{n=0}^{\infty} |x_{n+1} - x_n|$ が有界であるようなものとする．実数 $a, b \in \mathbb{R}$ に対して，$a \dot{-} b = \max\{a - b, 0\}$ によって定義する．このとき，次を考える．

$$y_n = x_0 + \sum_{i=0}^{n} (x_{i+1} \dot{-} x_i), \qquad z_n = \sum_{i=0}^{n} (x_i \dot{-} x_{i+1}).$$

明らかに両方とも非減少であり，$(x_n)_{n\in\omega}$ の選び方より，両方とも有界である．したがって，$y = \lim_{n\to\infty} y_n$ と $z = \lim_{n\to\infty} z_n$ は存在する．このとき，次が分かる．

$$\begin{aligned} y - z &= \lim_{n\to\infty} y_n - \lim_{n\to\infty} z_n = \lim_{n\to\infty} (y_n - z_n) \\ &= \lim_{n\to\infty} \left(x_0 + \sum_{i=0}^{n} (x_{i+1} \dot{-} x_i) - \sum_{i=0}^{n} (x_i \dot{-} x_{i+1}) \right) \\ &= \lim_{n\to\infty} \left(x_0 + \sum_{i=0}^{n} (x_{i+1} - x_i) \right) = \lim_{n\to\infty} x_n = x. \end{aligned}$$

(2)⇒(1): $(y_n)_{n\in\omega}$ と $(z_n)_{n\in\omega}$ をそれぞれ y と z に収束する有理数の計算可能上昇列とする．こ

のとき，$x_n = y_n - z_n$ と定義する．いま，$x = y - z$ とすれば，$x = \lim_n x_n$ である．さらに，

$$\begin{aligned}\sum_{n=0}^{\infty} |x_{n+1} - x_n| &= \sum_{n=0}^{\infty} |(y_{n+1} - z_{n+1}) - (y_n - z_n)| \\ &\leqslant \sum_{n=0}^{\infty} (y_{n+1} - y_n) + \sum_{n=0}^{\infty} (z_{n+1} - z_n) \\ &= \lim_{n\to\infty} (y_n - y_0) + \lim_{n\to\infty} (z_n - z_0) \\ &= y - y_0 + z - z_0 < \infty\end{aligned}$$

であるから，x は弱計算可能である． □

例 5.18. 下半計算可能でも上半計算可能でもない弱計算可能実数が存在する．

（ヒント：この証明には，**フリードバーグ・ムチニクの定理** (*Friedberg-Muchnik's theorem*) を利用する．これは，半計算可能集合 $A, B \subseteq \mathbb{N}$ で，$A \not\leqslant_T B$ かつ $B \not\leqslant_T A$ なるものが存在する，というものである．すると，$0.(A \oplus \overline{B}) = 0.(A \oplus \mathbb{N}) - 0.(\varnothing \oplus B)$ が求める実数となる．）

ここまでで見たように，極限計算可能実数にも様々なものがあり，以下のような包含関係が成り立っている．

計算可能 $\subsetneq$ 上半計算可能，下半計算可能 $\subsetneq$ 弱計算可能 $\subsetneq$ 2 進 ω-近似計算可能 $\subsetneq$ 極限計算可能

豆知識． 上半計算可能実数と下半計算可能実数はそれぞれ可換半環をなす．計算可能実数，弱計算可能実数，極限計算可能実数はそれぞれ実閉体をなす．

6 アルゴリズム情報理論

ランダムとは何なのか，ということを理解することは，「ランダムでない」とは何なのか，というのを理解することと同等である．そういうわけで，まずは，ランダムではない列について学んでいこう．以下に挙げるのは，明らかにランダムでないバイナリ列である．

$$\underbrace{0000000000000000000000000000\ldots\ldots00000000}_{1\text{ 億桁}},$$
$$\underbrace{01110111011101110111011101110111\ldots\ldots01110111}_{1\text{ 億桁}}.$$

上の列は，ただの「0 が 1 億個並ぶ列」であるから，明らかにランダムでない．下の列は，すぐには気付かないかもしれないが，よく見ると，0111 を繰り返しているだけであることが分かる．つまり，下の列とは「0111 が 2500 万回並ぶ列」であるから，ランダムではない．

それでは，これらのランダムでない列が共有する性質とは何であろうか．すぐに思いつくのは，これらの列が**規則的**である，という点であろうか．ランダム性のことを無秩序性や不規則性と呼ぶこともあるし，つまり，**規則性**というものは，ランダム性の対義語と考えられている．

それでは，そもそも規則的なバイナリ列とは何であろうか．その列を記述する法則か数式か論理式のようなものがある，という感じだろうか．ここでは，もう少し漠然と，

規則性とは，そのバイナリ列を「より短い言葉で説明できる」ということである

と考えよう．たとえば，上の例では，長さ 1 億のバイナリ列を「0 が 1 億個並ぶ列」という 8 文字，「0111 が 2500 万回並ぶ列」という 14 文字で説明することができた．もう少し身近な言葉で説明すると，これは，数十メガバイトの容量を持つデータをほんの数十バイトのデータに**圧縮**したということである．

以上をまとめると，非ランダム性とは規則性であり，規則性とは圧縮可能性である．言い換えれば，

$$\text{ランダム性} = \text{不規則性} = \text{圧縮不可能性}$$

ということである．この発想を，もう少し数学的に厳密な形で定式化しよう．

6.1 コルモゴロフ複雑性

先程はバイナリ列と言ったが，これはつまりアルファベット $\{0,1\}$ 上の有限列，つまり $\{0,1\}^*$ の要素のことである．よって，$\{0,1\}^*$ はバイナリ列全体の集合を表す．また，バイナリ列 $\sigma \in \{0,1\}^*$ の長さを $|\sigma|$ によって表す．

それではデータ圧縮の概念を数学的に定式化しよう．ここで考えるのは可逆圧縮であり，不可逆圧縮は考えない．可逆圧縮で重要なのは，解凍アルゴリズムである．圧縮されたデータから元のデータを復元できなければならない．上の例で言えば，「0 が 1 億個並ぶ列」という文字列が，実際に本物の 0 が 1 億個並ぶバイナリ列を意味しているという了解があるから，「0 が 1 億個並ぶバイナリ列は『0 が 1 億個並ぶ列』という文字列で説明された」と言えるのである．

つまり，σ という記述から τ というバイナリ列を実際に生成する方法 M があって初めて，τ は σ によって説明される，τ は σ に圧縮される，のように言うことができる．この M とは，部分計算可能関数 $M :\subseteq \{0,1\}^* \to \{0,1\}^*$ であり，$M(\sigma) = \tau$ のとき，τ は σ に圧縮された，と考える．そうすると，τ は M によってどれくらい小さいデータサイズまで圧縮できるか，の限界値は以下によって与えられる．

$$C_M(\tau) = \min\{|\sigma| : M(\sigma) = \tau\}.$$

この数値 $C_M(\tau)$ を τ の**平コルモゴロフ複雑性** (*plain Kolmogorov complexity*) と呼ぶ．もし $C_M(\tau) \geqslant |\tau|$ ならば，τ は決して圧縮できない列であり，したがって，ランダムな列と考えられる．まず，圧縮不可能列が存在することを見てみよう．

命題 6.1. どんな n についても，長さ n のバイナリ列 τ で，$C_M(\tau) \geqslant |\tau|$ を満たすものが存在する．

Proof. 長さ n 未満のバイナリ列の個数を数えよう．以下，$(\sigma)_2$ によって，σ を数を 2 進表記だと思ったときのその値を表す．たとえば，$(110)_2 = 6$ である．まず、長さ n のバイナリ列の種類は

ちょうど 2^n 個であることに注意しよう．すると，長さ n 未満のバイナリ列の個数は

$$\sum_{i=0}^{n-1} 2^i = (\underbrace{111\ldots111}_{n\text{ 個}})_2 = (1\underbrace{000\ldots000}_{n\text{ 個}})_2 - 1 = 2^n - 1$$

である．つまり，高々 $2^n - 1$ 個のバイナリ列 τ だけが $C_M(\tau) < n$ となり得る．しかし，長さ n のバイナリ列は 2^n 種類存在するから，長さ n のバイナリ列 τ で $C_M(\tau) \geqslant n$ となるようなものが存在する．つまり，$C_M(\tau) \geqslant |\tau|$ である． □

ところで，上では M は部分計算可能関数だったら何でも良いとしたが，以後は，入力文字列 σ がいつ読み込み終わるか分からないシチュエーションを考えたい．こういう状況では，M は，適当なタイミングで文字列の読み込みが終了したと判断して，出力を返す必要がある．たとえば，入力文字列側が自身のファイルサイズを最初に記述しているとか，あるいは入力文字列の最後にエンドマークが打たれているなどすれば，M はいつ文字列の読み込みが終了したかを判断できるであろう．ここでは，その判断方法は具体的には指定せず，単に，いつ終わるか分からない文字列を入力とする関数，という概念を以下のように定義する．

定義 6.2. 接頭機械 (*prefix-free machine*) とは，次を満たす部分計算可能関数 $M :\subseteq \{0,1\}^* \to \{0,1\}^*$ である．

$$(\forall \sigma, \tau \in \{0,1\}^*)\ [\sigma \prec \tau\ \&\ \sigma \in \mathrm{dom}(M) \implies \tau \notin \mathrm{dom}(M)].$$

つまり，M は何らかの文字列を読み込み中に，ある時点 σ で出力 $M(\sigma)$ を返したならば，その時点で文字列の読み込みは打ち切っており，σ の拡張 τ については M はもはや反応を示さない．

例 6.3. $\Phi :\subseteq \{0,1\}^* \to \{0,1\}^*$ を任意の部分計算可能関数とすると，

$$M(0^{|\sigma|}1\sigma) = \Phi(\sigma)$$

で定義される関数は接頭である．

定義 6.4. 接頭機械 R が**最適** (*optimal*) であるとは，任意の接頭マシン M に対して，次の条件を満たすものである．

$$(\exists c \in \mathbb{N})(\forall \tau \in \{0,1\}^*)\ C_R(\tau) \leqslant C_M(\tau) + c.$$

定理 6.5. 最適接頭機械は存在する．

Proof. 万能チューリング機械の存在より，$\Phi(0^e1\sigma) = [\![e]\!](\sigma)$ となるような部分計算可能関数 $\Phi :\subseteq \{0,1\}^* \to \{0,1\}^*$ が存在する．最適接頭機械 R の方針は以下である．入力 $0^e1\sigma$ に対して，$(\Phi(0^e1\tau) : \tau \sqsubseteq \sigma \text{ or } \sigma \sqsubseteq \tau)$ を同時にシミュレートし，その中で $\Phi(0^e1\sigma)$ の計算が停止するのが最速であったなら，$R(0^e1\sigma) = \Phi(0^e1\sigma)$ と出力する．より正確には，R を以下のように構成する．

入力 $0^e1\sigma$ に対して，$R(0^e1\sigma)$ の値を決めるために，まず $\Phi(0^e1\sigma)$ をシミュレートする．もし，ある s ステップで $\Phi(0^e1\sigma)$ が出力を返したならば，$s' = \max\{s, |\sigma|\}$ とする．各始切片 $\tau \sqsubset \sigma$ について，$\Phi(0^e1\tau)$ を s' ステップだけ実行し，その間に出力を返すかどうかを確かめる．もし $\Phi(0^e1\tau)$ が出力を返したならば，$R(0^e1\sigma)$ は出力を返さないと宣言する．さもなくば，続いて，各拡張 $\tau \sqsupset \sigma$ で $|\tau| < s$ なるものについて，$\Phi(0^e1\tau)$ を $s'-1$ ステップだけ実行し，その間に出力を返すかどうかを確かめる．もし $\Phi(0^e1\tau)$ が出力を返したならば，$R(0^e1\sigma)$ は出力を返さないと宣言する．さもなくば，$R(0^e1\sigma)$ は $\Phi(0^e1\sigma)$ を出力する．

いま，d を与えられた接頭機械 M のコードとする．このとき，$\Phi(0^d1\sigma) = M(\sigma)$ であるが，M は接頭機械であるから，与えられた σ について，$(\Phi(0^d1\tau) : \tau \sqsubseteq \sigma \text{ or } \sigma \sqsubseteq \tau)$ のうちで停止する $\Phi(0^d1\tau)$ は高々 1 つしか存在しない．したがって，$M(\sigma)$ が停止するならば，$R(0^d1\sigma)$ は停止し $\Phi(0^d1\sigma) = M(\sigma)$ を出力する．

いま，$C_M(\tau) \leqslant n$ とすると，長さ n 以下のバイナリ列 σ が存在して，$M(\sigma) = \tau$ となる．よって，$R(0^d1\sigma) = \tau$ を得るが，$|0^d1\sigma| \leqslant n + d + 1$ であるから，$C_R(\tau) \leqslant n + d + 1$ である．以上より，$C_R(\tau) \leqslant C_M(\tau) + d + 1$ が得られた． □

M が最適接頭機械であるとき，$C_M(\tau)$ の代わりに $K(\tau)$ と書き，これを**接頭コルモゴロフ複雑性** (*prefix-free Kolmogorov complexity*) と呼ぶ．以下，$\leqslant^+$ によって，高々定数 c 程度の差を除いて，不等式 $\leqslant$ が成立することを意味する．

命題 6.6. $K(\tau) \leqslant^+ 2|\tau|$.

Proof. $M(0^{|\tau|}1\tau) = \tau$ と定義すれば，これは接頭機械であり，$C_M(\tau) \leqslant 2|\tau| + 1$ を得る． □

自然数 $n \in \mathbb{N}$ が与えられたとき，$\mathsf{bin}(n)$ によって，n を 2 進展開したバイナリ列を表すものとする．このとき，$|\mathsf{bin}(n)| =^+ \log n$ である．ここで，対数関数 log の底は 2 である．

命題 6.7. $K(\tau) \leqslant^+ K(0^{|\tau|}) + |\tau| \leqslant^+ 2\log|\tau| + |\tau|$.

Proof. R を最適接頭機械とする．先程のように，文字列の長さ $|\tau|$ の情報をヘッダに付与するが，それを $0^{|\tau|}1$ と記述するのは無駄が多い．代わりに，文字列の長さ情報を圧縮したデータをヘッダに付与しよう．つまり，入力バイナリ列 η が与えられたとき，$R(\sigma) = \mathsf{bin}(|\tau|)$ となるような分解 $\eta = \sigma\tau$ を探し，そのようなものが見つかったら，$M(\sigma\tau) = \tau$ と定義する．このとき，M は接頭機械であり，$C_M(\tau) \leqslant C_R(0^{|\tau|}) + |\tau|$ である．補題 6.6 より，$C_R(0^{|\tau|}) \leqslant^+ 2\log|\tau|$ である． □

次の補題は，様々なバイナリ列のコルモゴロフ複雑性の上界を求める際に便利である．

補題 6.8. M が部分計算可能関数ならば，次を満たす定数 $c \in \mathbb{N}$ が存在する．

$$(\forall \tau \in \{0,1\}^*)\ K(M(\tau)) \leqslant K(\tau) + c.$$

Proof. R を最適接頭マシンとする．このとき，$S(\sigma) = M(R(\sigma))$ によって定義すると，$\mathrm{dom}(S) \subseteq \mathrm{dom}(R)$ であるから，S は接頭機械である．与えられた τ について，長さ $K(\tau)$ のバイナリ

列 σ で，$R(\sigma) = \tau$ なるものが存在する．このとき，$S(\sigma) = M(R(\sigma)) = M(\tau)$ であるから，$C_S(M(\tau)) \leqslant K(\tau)$ である．よって，R の最適性より，ある c が存在して，任意の τ に対して，$K(M(\tau)) \leqslant C_S(M(\tau)) + c \leqslant K(\tau) + c$ となる． □

自然数 $n \in \mathbb{N}$ について，$K(\mathsf{bin}(\mathrm{n}))$ のことを単に $K(n)$ と略記する．すると，K は自然数上の関数 $K : \mathbb{N} \to \mathbb{N}$ と思うことができる．この関数は非有界，つまり $\liminf_{n\to\infty} K(n) = \infty$ であるが，この発散速度はとてつもなく遅い．次の関数を考えよう．

$$K^-(n) = \min\{k \in \mathbb{N} : (\forall s \geqslant k)\ K(s) \geqslant n\}.$$

つまり，与えられた n に対して，$K^-(n)$ 以上の値で，コルモゴロフ複雑性が n 未満になるものは存在しない．さて，命題 3.5 において，どんな計算可能関数よりも急増大する関数，ビジービーバー関数について議論した．この関数 K^- もまた，以下のようにビジービーバー的な振る舞いを見せる．

命題 6.9. 任意の計算可能関数 $g : \mathbb{N} \to \mathbb{N}$ に対して，次が成立する．

$$(\exists s \in \mathbb{N})(\forall n \geqslant s)\ g(n) < K^-(n).$$

Proof. 無限個の n について $K^-(n) \leqslant g(n)$ なる計算可能関数 g が存在すると仮定する．$h(n) = g(2n)$ より定義すると，h も計算可能関数である．このとき，K^- の定義より，無限個の n について，$K(h(n)) \geqslant 2n$ である．一方，補題 6.8 より，ある定数 c が存在して，任意の n について $K(h(n)) \leqslant n + c$ となり，矛盾を導く． □

6.2 チャイティンのオメガ

それでは，具体的なランダム列を記述する準備を始めよう．接頭マシンにおいて，入力はいつ終わるか分からないバイナリ列であったことを思い出そう．いま，0 と 1 が同程度に出現する公平なコイン投げの繰り返しによってバイナリ列 $x_1x_2x_3\ldots$ を作る，という状況を考える．このような入力に対して，接頭機械 M が出力を返す確率 Ω_M はどれくらいだろうか？

たとえば，ある偶数 k について 0^k1 の形である入力のみ受理し，k を出力する接頭機械 M を考えよう．コインを $2n+1$ 回投げた結果が正確に $0^{2n}1$ となる確率は $0^{-(2n+1)}$ である．したがって，コイン投げを延々繰り返した結果，そのうち，ある偶数 k について 0^k1 の形となっている確率は以下によって与えられる．

$$\Omega_M = \sum_{n=0}^{\infty} 2^{-(2n+1)} = 2/3.$$

さて，どんなバイナリ列 $\sigma \in \{0,1\}^*$ が与えられても，コインを $|\sigma|$ 回投げた結果が正確に σ と一致する確率は $2^{-|\sigma|}$ である．したがって，一般の接頭機械 M についても，M が出力を返す確率 Ω_M は次によって計算できる．

定義 6.10 (停止確率)**.** M が接頭機械であるとき，M の**停止確率** (*halting probability*) とは，以下によって定まる値である．

$$\Omega_M = \sum_{\sigma \in \mathrm{dom}(M)} 2^{-|\sigma|}.$$

ここで，$\mathrm{dom}(M)$ は M の定義域を表す．最適接頭機械 R の停止確率 $\Omega = \Omega_R$ を特に**チャイティンの定数** (*Chaitin's constant*) と呼ぶ．

演習問題 6.11. M が接頭機械であるとき，$0 \leqslant \Omega_M \leqslant 1$ であることを証明せよ．

接頭機械 M が与えられたとき，停止確率 Ω_M は，計算可能な方法で下から近似できる．これを確かめるために，M_s を M の長さ s 以下の入力に対する計算を s ステップだけ実行したものとする．つまり，$|\sigma| \leqslant s$ かつ $M(\sigma)$ が s ステップ以下で出力を返すならば，$M_s(\sigma) = M(\sigma)$ とし，さもなくば $M_s(\sigma)$ は出力を返さないものとする．すると，

$$\Omega_{M,s} = \sum_{\sigma \in \mathrm{dom}(M_s)} 2^{-|\sigma|}$$

であり，この値は有限種類の入力に対する M の計算を s ステップずつ実行することによって計算できる．さらに，

$$\Omega_{M,0} \leqslant \Omega_{M,1} \leqslant \Omega_{M,2} \leqslant \cdots \leqslant \Omega_M = \lim_{s \to \infty} \Omega_{M,s}.$$

あるバイナリ列 $\sigma \in \{0,1\}^*$ について $\sum_{i=0}^{|\sigma|-1} \sigma(i) 2^{-i-1}$ と書けるような実数を**二進有理数** (*dyadic rational*) と呼ぶ．つまり，$0.\sigma 0^\omega$ の形の実数のことである．たとえば，任意の機械 M および $s \in \mathbb{N}$ について，$\mathrm{dom}(M_s)$ は有限であるから，$\Omega_{M,s}$ は二進有理数である．一方，R が最適接頭機械であれば，定義域は無限集合となり，したがって，二進有理数には成り得ない．つまり，Ω は二進有理数ではない．

演習問題 6.12. 最適接頭機械の定義域が無限集合であることを示し，したがって Ω が二進有理数でないことを結論づけよ．

二進有理数でないような実数 $x \in [0,1]$ の二進表記は一意に定まるので，x を無限バイナリ列と同一視できる．無限バイナリ列 $z \in \{0,1\}^{\mathbb{N}}$ が与えられたとき，$z \upharpoonright n$ によって，長さ n の z の始切片を表すものとする．つまり，$z = z_0 z_1 z_2 \ldots z_i z_{i+1} \ldots$ ならば，$z \upharpoonright n = z_0 z_1 z_2 \ldots z_{n-1}$ のことである．

定理 6.13. 任意の最適接頭機械の停止確率 Ω について，次が成立する．

$$(\exists c)(\forall n)\ K(\Omega \upharpoonright n) \geqslant n - c.$$

Proof. R を最適接頭機械とする．$\Omega = \Omega_R$ かつ $\Omega_s = \Omega_{R,s}$ とする．次のような（接頭とは限らな

い）部分計算可能関数 M を考える．各入力 σ に対して，$0.\sigma \leqslant \Omega_t < 0.\sigma + 2^{-n}$ になるまで待つ．もし，そのようなステップ t が訪れたら，R_t の値域に入っていないバイナリ列 η を返す．つまり，どんな σ についても $R_t(\sigma) \neq \eta$ であるような η を選び，$M(\sigma) = \eta$ とする．

さて，実数 x について，$\sigma = x \upharpoonright n$ であるとは，$0.\sigma \leqslant x \leqslant 0.\sigma + 2^{-n}$ であるということに注意しよう．したがって，もし $\sigma = \Omega \upharpoonright n$ ならば，$0.\sigma \leqslant \Omega \leqslant 0.\sigma + 2^{-n}$ である．$(\Omega_s)_{s\in\mathbb{N}}$ は非減少であり，$\Omega = \lim_{s\to\infty} \Omega_s$ であることと，$\Omega = 0.\sigma$ では有り得ないことを利用すると，$0.\sigma \leqslant \Omega_t < 0.\sigma + 2^{-n}$ となるような t が存在することが分かる．つまり，部分計算可能関数 M は入力 $\sigma = \Omega \upharpoonright n$ に対しては，必ず R_t の値域に入っていない η を出力する．

一方，t ステップより後では，長さ $n-1$ 以下の入力に対して R が出力を返すことはない．なぜなら，さもなくば，停止確率の定義より $\Omega \geqslant \Omega_t + 2^{-n+1}$ となるが，t の選び方より，$0.\sigma + 2^{-n+1} \leqslant \Omega_t + 2^{-n+1} \leqslant \Omega \leqslant 0.\sigma + 2^{-n}$ となり，これは矛盾を導く．

さて，η は R_t の値域に入ってなかったため，もし $R(\sigma) = \eta$ となるような σ があったとしても，その計算は t ステップよりも長い時間がかかる．したがって，上で見た t の性質より，R が入力 σ に対して出力を返しているということは，$|\sigma| \geqslant n$ を導く．これより，$K(\eta) \geqslant n$ となる．ここで $\eta = M(\sigma) = M(\Omega \upharpoonright n)$ だったことを思い出すと，$K(M(\Omega \upharpoonright n)) \geqslant n$ を得る．ところで，M は部分計算可能関数であるから，補題 6.8 より，任意の $\sigma \in \{0,1\}^*$ に対して，$K(M(\sigma)) \leqslant K(\sigma) + c$ である．以上をまとめると，

$$n \leqslant K(M(\Omega \upharpoonright n)) \leqslant K(\Omega \upharpoonright n) + c$$

である．ここで，c は部分計算可能関数 M に依存する定数であって，n に依存しないことに注意する．以上より，任意の n について，$K(\Omega \upharpoonright n) > n - c$ を得る． □

6.3 マーティンレフ・ランダムネス

さて，最初に，ランダム性とは圧縮不可能性であると述べた．しかし，多くの人々の印象としては，ランダム性とは確率概念と大きく結びつくものである．したがって，ランダム性と圧縮不可能性の “同値性” をもう少し確率論的に理解する方法はないだろうか．

ランダム系列であれば，確率 100% で成り立つ法則を全て満たしているべきである．たとえば，ランダム系列は，ボレルの**大数の法則**を満たすし，ヒンチンの**重複対数の法則**も満たす．このような確率 100% で成立する法則を全て満たすような無限列であれば，誰もがそれをランダムであると承認するのではないだろうか．しかし，如何なる無限列 $\alpha \in \{0,1\}^{\mathbb{N}}$ に対しても，「α である確率」は 0% である．つまり，ランダムな無限列は決して α には為りえない．—**確率論的にランダムな無限列は存在しない**！

この非存在証明は数学的には正しいが，計算論的観点からすると超越的である．そもそも我々はこの無限列 α が何であるかを知らないから，如何にして事前に α について語ることが出来ようか．無限列 α を知らずして，「α である」などと一体どのように言及するのか．しかし，**有限の記述** e があって，それが何らかの意味で無限列 α_e を表すのであれば，「『α_e である確率』は 0% である」

という言明は，我々は実際に有限の言葉で語ることができる．このように，有限を無限に変換する規則 $e \mapsto \alpha_e$ があるならば，「α_e である」という言及は実際に可能であるし，許されるべきだろう．したがって，ランダム性の定義を次のように改良しよう．

> ランダム系列であれば，確率 100% で成り立つような**有限の言葉で記述できる**法則を全て満たしているべきである．

ボレルの大数の法則も，ヒンチンの重複対数の法則も，有限の言葉で記述することができる．とはいえ，「有限の言葉で記述できる」とはどういう意味なのか曖昧であるから，これでは定義になっていない．マーティンレフが提案したことは，ランダム性概念とは，アルゴリズム概念を介して言及可能な確率 0% の条件を決して発生させないものである．

以下，正確な定義を与えよう．バイナリ列の集合 $U \subseteq \{0,1\}^*$ が与えられているとしよう．このとき，U の**確率** (*probability*) または**測度** (*measure*) を，表と裏の出現確率が等しいコイン投げの試行の繰り返しによって得られるバイナリ列がいつか U に含まれる確率によって定義し，$\lambda(U)$ と書く．数学的には，U の確率 $\lambda(U)$ は以下の式によって与えられる．

$$\lambda(U) = \sum\{2^{-|\sigma|} : \sigma \in U \text{ and } (\forall \tau \sqsubseteq \sigma)\ \tau \notin U\}.$$

注意. λ は，公平なコイン投げから得られる $\{0,1\}^{\mathbb{N}}$ 上の確率測度 λ' と同一視できる．具体的には，バイナリ列 $\sigma \in \{0,1\}^*$ に対して，$[\sigma]$ を σ を拡張する無限バイナリ列全体，つまり $[\sigma] = \{\alpha \in \{0,1\}^{\mathbb{N}} : \sigma \sqsubseteq \alpha\}$ によって定義する．また，$[U]$ を U が生成する開集合 $[U] = \bigcup_{\sigma \in U}[\sigma]$ とする．このとき，$\lambda(U) = \lambda'([U])$ であることが分かる．

$\{0,1\}^*$ の部分集合の列 $(U_n)_{n\in\mathbb{N}}$ が与えられているとする．この集合列が**計算可枚挙**であるとは，$\{(n,\sigma) \in \mathbb{N} \times \{0,1\}^* : \sigma \in U_n\}$ が計算可枚挙であることを意味する．

定義 6.14. 集合 $N \subseteq \{0,1\}^n$ が**マーティンレフ零**であるとは，$N \subseteq \bigcap_{n\in\mathbb{N}}[U_n]$ なる Σ^0_1 集合列 $\{U_n\}_{n\in\mathbb{N}}$ で，任意の $n \in \mathbb{N}$ について $\lambda(U_n) \leqslant 2^{-n}$ なるものが存在するときを言う．このとき，無限列 $\alpha \in \{0,1\}^{\mathbb{N}}$ が**マーティンレフ・ランダム** (*Martin-Löf random*) であるとは，$\{\alpha\}$ がマーティンレフ零集合でないときを指す．

上の定義の中の集合列 $(U_n)_{n\in\mathbb{N}}$ は，**マーティンレフ検定** (*Martin-Löf test*) と呼ばれる．これが意味するものは，アルゴリズム的に記述された統計的検定である．つまり，半計算可能な方法で記述された集合であり，確率 0 となることが半計算可能な方法で保証されている．無限列 $\alpha \in \{0,1\}^{\mathbb{N}}$ がマーティンレフ・ランダムであることは，どんなマーティンレフ検定 $(U_n)_{n\in\mathbb{N}}$ に対しても，$\alpha \notin \bigcap_n U_n$ となることと等しい．

さて，この統計的検定による確率的ランダム性と，圧縮不可能性によるランダム性には如何なる関わりがあるだろうか．実は，確率的にランダム性を定義しても圧縮不可能性によってランダム性を定義しても同値であることを数学的に証明できる，というのが次の定理の述べるところである．

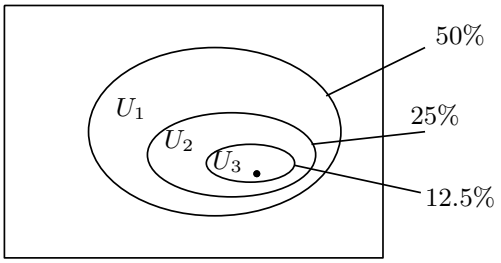


図 3　ランダムでないならば，マーティンレフ零集合によって捕捉される．

定理 6.15. 無限バイナリ列 $\alpha \in \{0,1\}^{\mathbb{N}}$ に対して，以下の 2 条件は同値である．

1. α はマーティンレフ・ランダムである．
2. 次を満たす定数 $c \in \mathbb{N}$ が存在する．

$$(\forall n \in \mathbb{N})\ K(\alpha \upharpoonright n) \geqslant n - c.$$

このために，以下の補題が必要である．集合 $L \subseteq \{0,1\}^* \times \mathbb{N}$ について，$\sum_{(\sigma,r)\in L} 2^{-r} \leqslant 1$ を**クラフトの不等式** (*Kraft's inequality*) と呼ぶ．

補題 6.16 (機械存在補題)**.** 計算可枚挙集合 $L \subseteq \{0,1\}^* \times \mathbb{N}$ がクラフトの不等式を満たすとする．このとき，ある部分計算可能接頭関数 M が存在して，全ての $(\sigma, r) \in L$ について，長さ r のある記述 $\tau \in \{0,1\}^r$ によって，$M(\tau) = \sigma$ が成立する．特に，$K_M(\sigma) \leqslant r$ を満たす．

Proof. $L = \{(\sigma_i, r_i)\}_{i\in\mathbb{N}}$ と計算可能に枚挙する．反鎖 $\{\tau_n\}_{n\in\mathbb{N}}$ で，各 $n \in \mathbb{N}$ について $|\tau_n| = r_n$ なるものを計算可能な手法で構成する．各 $n \in \mathbb{N}$ について，$\xi_n \in \{0,1\}^*$ をその 2 進表現が次の条件を満たすものとして定義する．

$$0.\xi_n = 1 - \sum_{j=0}^{n} 2^{-r_j}.$$

これから，$\xi_n(m) = 1$ なる各 $m \in \mathbb{N}$ について，補助的な有限列 $\rho_{n,m} \in \{0,1\}^{m+1}$ を定義する．これは，M の構成の第 n 段階時点での $\mathrm{dom}(M)$ の利用可能領域を表し，ある $m \in \mathbb{N}$ について，$\tau_{n+1} \succeq \rho_{n,m}$ として定義される．このとき，常に $\{\tau_k : k \leqslant n\} \cup \{\rho_{n,m} : \xi_n(m) = 1\}$ が反鎖となるように定義されるならば，結果として $\mathrm{dom}(M)$ が反鎖となることが示される．

構成の第 1 段階として，$\tau_0 = 0^{r_0}$ によって定義する．このとき，次が分かる．

$$2^{-r_0} = 0.\underbrace{000\ldots000}_{r_0-1}1,$$
$$0.\xi_0 = 0.\underbrace{111\ldots111}_{r_0-1}.$$

各 $m < r_0$ について，$\rho_{0,m} = 0^m 1$ によって定義する．$\{\tau_0\} \cup \{\rho_{0,m} : \xi_0(m) = 1\}$ が反鎖であることは明らかである．いま，反鎖 $\{\tau_k : k \leqslant n\} \cup \{\rho_{n,m} : \xi_n(m) = 1\}$ が既に構成されていると仮定する．

r_{n+1} が与えられたとき，$0.\xi_{n+1} = 0.\xi_n - 2^{-r_{n+1}}$ なので，もし，$\xi_n(r_{n+1} - 1) = 1$ ならば，ξ_{n+1} は ξ_n の第 $r_{n+1} - 1$ 桁目を 0 に切り替えたものである．つまり，

$$\xi_n = \langle x_0, x_1, x_2, x_3, \ldots, x_{r_{n+1}-2},\ \mathbf{1}\ , x_{r_{n+1}}, \ldots \rangle,$$
$$\xi_{n+1} = \langle x_0, x_1, x_2, x_3, \ldots, x_{r_{n+1}-2},\ \mathbf{0}\ , x_{r_{n+1}}, \ldots \rangle.$$

である．このとき，$\tau_{n+1} = \rho_{n,r_{n+1}-1}$ とする．

一方，$\xi_n(r_{n+1} - 1) = 0$ ならば，$\xi_n(j) = 1$ なる最大の $j < r_{n+1}$ に対して，ξ_n の第 j 桁目を 0 に，$j + 1$ 以上 r_{n+1} 未満の桁を 1 に切り替えたものが ξ_{n+1} である．つまり，

$$\xi_n = \langle x_0, x_1, \ldots, x_{j-1},\ \mathbf{1}\ , 0, 0, 0, \ldots, 0, 0, 0,\ \mathbf{0}\ , x_{r_{n+1}}, \ldots \rangle,$$
$$\xi_{n+1} = \langle x_0, x_1, \ldots, x_{j-1},\ \mathbf{0}\ , 1, 1, 1, \ldots, 1, 1, 1,\ 1\ , x_{r_{n+1}}, \ldots \rangle.$$

となる．このとき，$\tau_{n+1} = \rho_{n,j}^\frown 0^{r_{n+1}-j}$ と定義し，各自然数 $m \in [j, r_{n+1})$ に対して，$\rho_{n+1,m} = \rho_{n,j}^\frown 0^{m-j} 1$ と定義する．$\{\tau_k : k \leqslant n + 1\} \cup \{\rho_{n+1,m} : \xi_{n+1}(m) = 1\}$ が反鎖となっていることは明らかである．

計算可能接頭関数 M を $M(\tau_n) = \sigma_n$ によって定義する．このとき，$\mathrm{dom}(M)$ は反鎖である．加えて，$|\tau_n| = r_n$ であることから，$K_M(\sigma) \leqslant r_n$ であることが従う． □

*Proof (*定理 6.15*).* $\neg(1) \Rightarrow \neg(2)$: 仮定より，ある計算可枚挙集合列 $\{U_n\}_{n\in\mathbb{N}}$ で，$\alpha \in \bigcap_n [U_n]$ かつ $\lambda(U_n) \leqslant 2^{-n}$ なるものが存在する．ある計算可枚挙な反鎖 $S_n \subseteq \{0,1\}^*$ で，$[U_n] = [S_n]$ なるものを見つけるのは難しくない．このとき，$\sum_{\sigma\in S_n} 2^{-|\sigma|} = \lambda(U_n) \leqslant 2^{-n}$ であることに注意する．後は $L = \{(\sigma, |\sigma| - n + 1) : \sigma \in S_{2n}\}$ と定義すれば，クラフトの不等式を満たす：

$$\sum_{(\sigma,r)\in L} 2^{-r} = \sum_{n=0}^{\infty} \sum_{\sigma\in S_{2n}} 2^{-(|\sigma|-n+1)} = \sum_{n=0}^{\infty} 2^{n-1} \cdot \lambda(U_{2n}) \leqslant \sum_{n=0}^{\infty} 2^{n-1} \cdot 2^{-2n} = 1.$$

よって，機械存在補題 6.16 より，ある接頭関数 M が存在して，任意の $\sigma \in S_{2n}$ について，$K_M(\sigma) \leqslant |\sigma| - n + 1$ となる．このとき，$\alpha \in \bigcap_t [S_t]$ なので，任意の t について，$\alpha \upharpoonright n_t \in S_{2t}$ となる $n_t \in \mathbb{N}$ が存在する．このとき，$K_M(\alpha \upharpoonright n_t) \leqslant n_t - t + 1$ であるから，(2) の式の否定が得られた．

$\neg(2) \Rightarrow \neg(1)$: 各 $c \in \mathbb{N}$ について，次の集合 V_c を考える．

$$V_c = \{\sigma \in \{0,1\}^{<\mathbb{N}} : K(\sigma) \leqslant |\sigma| - c\}.$$

仮定より，$\alpha \in \bigcap_c [V_c]$ であるため，$\lambda(V_c) \leqslant 2^{-c}$ であることを示せば十分である．R を最適接頭機械とする．もし $\sigma \in V_c$ ならば，それを記述する有限列 $p(\sigma) \in \{0,1\}^*$ で，$|p(\sigma)| \leqslant |\sigma| - c$ かつ $R(p(\sigma)) = \sigma$ を満たすものが存在する．前者より，$\sigma \in V_c$ について，測度に関する以下の不等式を得る．

$$\lambda([p(\sigma)]) \geqslant 2^{|\sigma|+c} = 2^c \cdot \lambda([\sigma]).$$

また，後者より，$\sigma \in V_c$ ならば $p(\sigma) \in \mathrm{dom}(R)$ である．よって，以下の不等式が導かれる．

$$2^c \cdot \lambda([V_c]) \leqslant 2^c \cdot \sum_{\sigma \in V_c} \lambda([\sigma]) \leqslant \sum_{\sigma \in V_c} \lambda([p(\sigma)]) \leqslant \sum_{\sigma \in \mathrm{dom}(R)} \lambda([\sigma]) = \Omega \leqslant 1.$$

これより，$\lambda([V_c]) \leqslant 2^{-c}$ を得る． □

6.4 ベルヌーイ測度とマルチンゲール

ここに，2 頭の馬 A と B がいるとしよう．この馬たち A と B が競争したとき，今の所 A の勝率が 70% であり，B の勝率が 30% であった．さて，カジノ運営者 $\mathcal{C}$ 氏は，この二頭の馬を用いた競馬賭博を開催することにした．このとき，各馬券の配当額を幾らに指定すれば，公平な賭博になるだろうか．

勝率に偏りのある競馬，あるいは表裏の出現確率に偏りのあるコインが生み出す確率測度は，ベルヌーイ測度と呼ばれる．ここでは，より一般の概念として，第 n 回目の試行では，表の出現確率が p_n であり，裏の出現確率が $1 - p_n$ であるようなコイン投げから得られる確率測度としてのベルヌーイ測度を導入する．以下，$\langle\rangle$ で空列を表す．

定義 6.17 (ベルヌーイ測度)**.** 実数列 $p = (p_n)_{n \in \mathbb{N}} \in [0,1]^{\mathbb{N}}$ が与えられたとき，次の 3 条件を満たす $m_p : \{0,1\}^* \to [0,1]$ から得られる $\{0,1\}^{\mathbb{N}}$ 上の確率測度 λ_p を**バイアス p のベルヌーイ測度** (*Bernoulli measure with bias p*) と呼ぶ：

1. $m_p(\langle\rangle) = 1$ である．
2. 任意の $\sigma \in \{0,1\}^n$ に対して，$m_p(\sigma 0) = p_n \cdot m_p(\sigma)$ である．
3. 任意の $\sigma \in \{0,1\}^n$ に対して，$m_p(\sigma 1) = (1 - p_n) \cdot m_p(\sigma)$ である．

例 6.18. 実数 $p \in [0,1]$ のみからなる実数列 $(p, p, p, \dots)$ を単に p で表す．λ_p をバイアス p のベルヌーイ測度とし，$\#i(\sigma)$ によって $\sigma \in \{0,1\}^*$ 中に現れる $i \in \{0,1\}$ の総数，つまり $\#\{n < |\sigma| : \sigma(n) = i\}$ を意味するものとする．このとき，各 $\sigma \in \{0,1\}^*$ について，$[\sigma]$ の λ_p-確率は以下の値となる：

$$\lambda_p([\sigma]) = m_p(\sigma) = p^{\#0(\sigma)} \cdot (1-p)^{\#1(\sigma)}.$$

より一般に，$\{0,1\}^{\mathbb{N}}$ 上のどんなボレル確率測度も，それぞれの有限的条件 $[\sigma]$ が成立する確率 $m(\sigma)$ がどの程度であるかを指定することによって得られる．

定理 6.19 (カラテオドリの拡張定理)**.** $m : \{0,1\}^* \to [0,1]$ を，$m(\langle\rangle) = 1$ かつ，任意の $\sigma \in \{0,1\}^*$ について $m(\sigma) = m(\sigma 0) + m(\sigma 1)$ となる関数とする．このとき，$\{0,1\}^*$ 上のボレル確率測度 μ_m で，任意の $\sigma \in \{0,1\}^*$ について $\mu_m([\sigma]) = m(\sigma)$ を満たすようなものが一意に存在する．

さて，0 の出現確率が p であり，1 の出現確率が $1-p$ であるとしよう．すると，オッズとしては，0 の出現を当てた場合には賭け金の $1/p$ 倍，1 の出現を当てた場合には賭け金の $1/(1-p)$ 倍の配当金が妥当であるように思える．ある賭博戦略が与えられているとして，コイン投げの繰り返しの結果の系列が σ であった場合の現在所持金額を $d(\sigma)$ によって表すものとする．たとえば，もし，0 が出現することに q_0 ドル，1 が出現することに q_1 ドル賭けた場合，資金の変動過程は，以下のようになる．

$$d(\sigma 0) = d(\sigma) - (q_0 + q_1) + \frac{q_0}{p},$$
$$d(\sigma 1) = d(\sigma) - (q_0 + q_1) + \frac{q_1}{1-p}.$$

したがって，$p \times$ (上式) $+ (1-p) \times$ (下式) を計算することによって，

$$d(\sigma) = pd(\sigma 0) + (1-p)d(\sigma 1)$$

という条件が満たされる．一般に，確率測度 μ が与えられており，現在までに得た 0 と 1 の列が σ であるとき，次に $i \in \{0,1\}$ が出現する確率は，条件付確率 $\mu([\sigma i]|[\sigma]) = \mu([\sigma i])/\mu([\sigma])$ によって得られる．

J. Ville は，賭博戦略の資金の変動過程として，マルチンゲールを定義した．つまり，「コイン投げの繰り返しの結果が σ であった時点での資金は $d(\sigma)$ である」ということを表す関数 $d : \{0,1\}^* \to [0,\infty)$ が Ville の意味でのマルチンゲールである．このようなマルチンゲールは，次のように形式的定義を与えることができる．

定義 6.20. μ を $\{0,1\}^{\mathbb{N}}$ 上の任意の確率測度とする．このとき，$d : \{0,1\}^* \to [0,\infty)$ が任意の $\sigma \in \{0,1\}^*$ に対して次の条件を満たすとき，μ-**マルチンゲール** (μ-*martingale*) と呼ばれる：

$$d(\sigma) = \frac{\mu([\sigma 0])d(\sigma 0) + \mu([\sigma 1])d(\sigma 1)}{\mu([\sigma])}.$$

ここで，d の値域は 0 以上の実数であることに注意する．つまり，我々が考える賭博戦略は，負債を負わないことを前提としている．

ある賭博師 G 氏は，1 ドルを元手に，どうにか 100 万ドル以上の大金を掴んで億万長者になりたいと考えている．このために，G 氏はある戦略 d を頼りに，コイン投げ賭博に挑むことにした．ただし，G 氏は，現在の持ち金が 100 万ドルを超えた瞬間に，それ以上欲張らずにゲームを打ち切るとしよう．さて，G 氏が目的を成就できる確率はどれくらいだろうか？

これに対する答えは，Ville の定理として知られる．如何なる戦略を用いようとも，元手となる初期資金が有限である限り，一度も借金をせずに所持金を 100 万倍に増やせる確率は，ほんの 100 万分の 1 である．つまりは，（公平なルールでゲームをする限りは）期待値よりも儲かる戦略なんてものは世の中には存在しない．

定理 6.21. μ を $\{0,1\}^{\mathbb{N}}$ 上の任意の確率測度とし，$q \geqslant 1$ を実数とする．d が μ-マルチンゲールならば，ある $n \in \mathbb{N}$ で $d(\alpha \upharpoonright n) \geqslant q \cdot d(\langle\rangle)$ となる α 全体の集合の μ-測度は q^{-1} 以下である．

Proof. 単純のために初期資金は $d(\langle\rangle) = 1$ であるとする．まず，S を $d(\sigma) \geqslant q$ を満たす極小な $\sigma \in \{0,1\}^*$ たちの集合として定義する．目標は，$\lambda([S]) \leqslant q^{-1}$ を示すことである．このとき，

$$\mu([S]) \cdot q = \sum_{\sigma \in S} \mu([\sigma]) \cdot q \leqslant \sum_{\sigma \in S} \mu([\sigma]) d(\sigma).$$

あとは，$\sum_{\sigma \in S} \mu([\sigma]) d(\sigma) \leqslant d(\langle\rangle) = 1$ であることを示せばよい．任意の数 $k \in \mathbb{N}$ について，$S[k] = S \cap \{0,1\}^{<k}$ と書く．もし，任意の $k \in \mathbb{N}$ について $a_k = \sum_{\sigma \in S[k]} \mu([\sigma]) d(\sigma) \leqslant 1$ ならば，$\lim_k a_k \leqslant 1$ であるから，以下を示せば十分であることが分かる．

主張. $\tau \in \{0,1\}^*$ を任意の有限列とする．もし $F \subseteq \{0,1\}^*$ が τ を拡張する有限列たちからなる有限 $\sqsubseteq$-反鎖ならば，$\sum_{\sigma \in F} \mu([\sigma]) d(\sigma) \leqslant \mu([\tau]) d(\tau)$ が成立する．

F の濃度に関する帰納法によって示す．$\#F = 1$ であるときは，たとえ τ 時点での所持金 $d(\tau)$ を元手に，以後の結果が $\sigma \sqsupseteq \tau$ のように続くという確信の下で全額賭け続けたとしても，$d(\sigma) \leqslant (\mu([\tau])/\mu([\sigma])) d(\tau)$ となることから，$\mu([\sigma]) d(\sigma) \leqslant \mu([\tau]) d(\tau)$ となり，主張は導かれる．

$\#F = n$ のとき主張は成立していると仮定する．F を濃度 $n+1$ の反鎖とする．τ を $F \subseteq \{\sigma : \tau \sqsubseteq \sigma\}$ なる最大の長さの有限列とする．このとき，各 $i < 2$ について $F_i = \{\sigma \in F : \tau i \sqsubseteq \sigma\}$ は濃度 n 以下である．帰納的仮定より，以下が導かれる．

$$\begin{aligned}\sum_{\sigma \in F} \frac{\mu([\sigma])}{\mu([\tau])} d(\sigma) = \sum_{i<2} \sum_{\sigma \in F_i} \frac{\mu([\sigma])}{\mu([\tau])} d(\sigma) &= \sum_{i<2} \sum_{\sigma \in F_i} \frac{\mu([\tau i])}{\mu([\tau])} \cdot \frac{\mu([\sigma])}{\mu([\tau i])} d(\sigma) \\ &\leqslant \sum_{i<2} \frac{\mu([\tau i])}{\mu([\tau])} d(\tau i) = d(\tau).\end{aligned}$$

両辺に $\mu([\tau])$ を掛けることによって，$\sum_{\sigma \in F} \mu([\sigma]) d(\sigma) \leqslant \mu([\tau]) d(\tau)$ を得る． □

特に，如何なる戦略を用いようとも，元手となる初期資金が有限である限り，一度も借金をせずに “所持金を好きなだけ増やせる確率” は，0% であることが分かる．

定理 6.22 (ボレル測度の正則性)**.** μ を $\{0,1\}^{\mathbb{N}}$ 上のボレル測度とする．このとき，任意の μ-可測集合 $A \subseteq \{0,1\}^{\mathbb{N}}$ に対して，開集合の下降列 $\{U_n\}_{n \in \mathbb{N}}$ で，次を満たすものが存在する：

$$\mu(A) = \inf_{n \to \infty} \mu(U_n), \text{ かつ } A \subseteq U_n.$$

$\{0,1\}^{\mathbb{N}}$ 上のボレル測度の正則性より，零集合を取り扱う際，測度が 0 に収束する開集合の下降列のみを考慮すればよい．以下，$\sigma \mapsto \mu([\sigma])$ が計算可能であるような測度 μ を**計算可能測度** (*computable measure*) と呼ぶ．

定義 6.23 (任意の測度に対するランダムネス)**.** μ を $\{0,1\}^{\mathbb{N}}$ 上の任意の計算可能確率測度とする．集合 $N \subseteq \{0,1\}^{\mathbb{N}}$ が**実効 μ-零** (*effectively μ-null*) であるとは，ある計算的枚挙可能集合列 $\{U_n\}_{n\in\mathbb{N}}$ で，任意の $n \in \mathbb{N}$ に対して，次を満たすものが存在するときを指す：

$$\mu(U_n) \leqslant 2^{-n}, \text{ かつ } N \subseteq U_n.$$

無限列 $\alpha \in \{0,1\}^{\mathbb{N}}$ に対して，$\{\alpha\}$ が実効 μ-零でないとき，α は**マーティンレフ μ-ランダム** (*Martin-Löf μ-random*) あるいは単に **μ-ランダム**であると呼ばれる．

さて，無限バイナリ列がランダムでない，ということは規則的である，ということであり，つまりバイナリ列の各ビットを予測できるということであった．予測可能性は，賭博によって儲けられる，ということに相当する．したがって，マルチンゲールとランダム性には深い関連性があることは想像に難くない．実際，以下によって，マルチンゲールと統計的ランダム性は数学的に厳密に結び付けられる．

定理 6.24. μ を $\{0,1\}^{\mathbb{N}}$ 上の任意の計算可能確率測度とする．このとき，任意の無限列 $\alpha \in \{0,1\}^{\mathbb{N}}$ に対して，次の 2 条件は同値である．

1. α は μ-ランダムである．
2. 任意の下半計算可能 μ-マルチンゲール $d : \{0,1\}^* \to [0,\infty)$ に対して，次の条件が成立する：
$$\limsup_{n\to\infty} d(\alpha \upharpoonright n) < \infty.$$

Proof. $\neg(1) \Rightarrow \neg(2)$: もし α が μ-ランダムでないならば，ある計算可能開集合の下降列 $\{U_n\}_{n\in\mathbb{N}}$ で，$\mu(U_n) \leqslant 2^{-n}$ かつ $\alpha \in \bigcap_n U_n$ なるものが存在する．初期資産 1 のうち 2^{-n} を元手にしたマルチンゲール d_n を以下の条件付確率によって定義する．

$$d_n(\sigma) = \mu(U_n|[\sigma]) = \frac{\mu(U_n \cap [\sigma])}{\mu([\sigma])}.$$

明らかに，$d_n(\langle\rangle) \leqslant 2^{-n}$ かつ $U_n \subseteq \{\beta \in \{0,1\}^{\mathbb{N}} : \lim_n d_n(\beta \upharpoonright n) = 1\}$ である．さらに，

$$\mu([\sigma])d_n(\sigma) = \mu(U_n \cap [\sigma]) = \sum_{i=0}^{1} \mu(U_n \cap [\sigma i]) = \sum_{i=0}^{1} \mu([\sigma i])d_n(\sigma i).$$

よって，$d = \sum_{n=0}^{\infty} d_n$ は下半計算可能マルチンゲールであり，以下の性質を満たす．

$$\alpha \in \bigcap_n U_n \subseteq \{\beta \in \{0,1\}^{\mathbb{N}} : \lim_n d(\beta \upharpoonright n) = \infty\}.$$

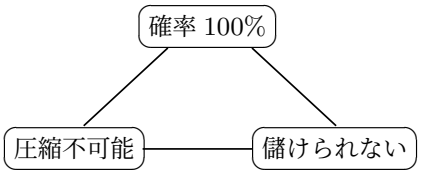


図 4　様々な発想に基づくランダム性概念は，実際には三位一体であり，一つの概念を別の視点に投影したものに他ならない (Schnorr 1971, 1973)

¬(2)⇒ ¬(1): d を下半計算可能マルチンゲールとする．U_n を次によって定義される開集合とする．

$$U_n = \{\alpha \in \{0,1\}^{\mathbb{N}} : (\exists n \in \mathbb{N})\ d(\alpha \upharpoonright n) \geqslant 2^n\}.$$

明らかに，次が満たされる．

$$\bigcap_n U_n = \{\alpha \in \{0,1\}^{\mathbb{N}} : \limsup_n d(\alpha \upharpoonright n) = \infty\}.$$

定理 6.21 によって，$\lambda(U_n) \leqslant 2^{-n}$ であるから，$\bigcap_n U_n$ の λ-測度が 0 であることが導かれる．　□

6.5　ハウスドルフ次元と複雑性

定理 6.24 によれば，統計的ランダム性（マーティンレフ・ランダム性）とマルチンゲールによるランダム性は，一般の確率測度においても一致する．それでは圧縮不可能性によるランダム性についてはどうだろうか．アルゴリズム情報理論としては，データ圧縮の性質を取り扱いたいため，コルモゴロフ複雑性の定義を確率測度毎に変えるのは望ましくない．したがって，素の状態のコルモゴロフ複雑性を用いて μ-ランダム列を調査したい．コルモゴロフ複雑性は公平なコイン投げによる確率測度 λ におけるランダム性との対応があったから，これは μ-ランダム列の確率測度 λ の下での振る舞いを考えるということである．

たとえば，μ として，バイアス $p = (p, p, p, \dots)$ のベルヌーイ測度 λ_p を考えよう．このとき，もちろん λ_p-ランダム列における 0 と 1 の出現確率は $p : (1-p)$ である．ところが，大数の法則から，100% の λ-確率で，0 と 1 の極限的な出現頻度は 1/2 になる．すると，0 と 1 の出現頻度が偏った無限列が得られる確率は 0% ということになる．それなら，出現頻度が $p : (1-p)$ の無限列は，0% のうちのどれくらいの量を占めるだろうか？　どうにかして「0% 以下の確率」の現象を調べる方法はあるだろうか？

長さ 1 の線分の 2 次元世界における “大きさ” は 0 であるが，より低い次元の目で見れば，つまり，1 次元世界における “大きさ” は 1 である．測度 0 のものを調べるには，低次元への移行が有効であるようだ．しかし，離散的な次元では，それぞれの次元間のギャップが大きすぎるため，様々な現象の精密な分析を行うには大味すぎる．そこで，整数次元のルベーグ測度 0 より精密な物差しとして，フラクタル幾何学などで頻繁に利用される，非整数次元のハウスドルフ測度の概念を持ち出すのがよい．この発想は，確率測度においても応用可能である．公平なコインによる確率測

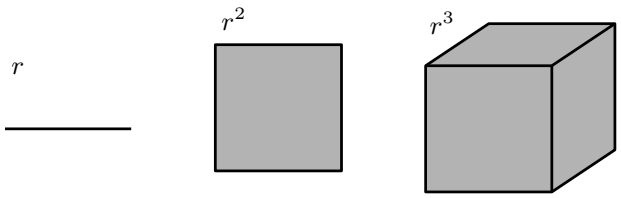


図 5　1 辺が r の s 次元立方体の s 次元ルベーグ測度は r^s である．

度 λ を 1 次元の確率測度と見立てよう．1 次元の確率論の代わりに今より展開するものは，**非整数次元の確率論**である．

定義 6.25 (ハウスドルフ測度 1917)**.** 実数 $s \in [0,1]$ を固定する．$A \subseteq \{0,1\}^{\mathbb{N}}$ の s **次元ハウスドルフ外測度** (*s dimensional outer Hausdorff measure*) は，以下によって与えられる．

$$\lambda^s(A) = \lim_{n\to\infty} \inf \left\{ \sum_{\sigma\in S} 2^{-s|\sigma|} : A \subseteq [S], \text{ and } S \subseteq \{0,1\}^{\geqslant n} \right\}.$$

注意. 定義における $2^{-s|\sigma|}$ は $(\lambda([\sigma]))^s$ に等しいことに注意する．s 次元ハウスドルフ測度の定義の発想は，スケール則に基づく．一辺が r の線分の長さは r であり，一辺が r の正方形の面積は r^2 であり，一辺が r の立方体の体積は r^3 である (図 5)．より一般に，$s \in \mathbb{N}$ について，1 辺が r の s 次元立方体の s 次元ルベーグ測度は r^s である．一般の非整数 $s \in [0,\infty)$ の次元についても，図形の直径とサイズには s 乗の相関があると想定することにより，定義 6.25 が正当化される．

命題 6.26. もし $A \subseteq \{0,1\}^{\mathbb{N}}$ が λ-可測ならば，$\lambda(A) = \lambda^1(A)$ である．

さて，長さ 1 の線分は，1 次元の世界では大きさ 1 を持つが，2 次元世界では大きさを認識できない，すなわち面積 0 であると言えるだろう．あるいは，一辺の長さ 1 の正方形は，面積 1 であるが，体積 0 であり，さらに，正方形を充填するには長さ ∞ の曲線が必要であるから，正方形は長さ ∞ であると考えられる．このように，ある図形が，本来あるべき次元より大きい次元にあるとき，その大きさは 0 であると認識され，本来あるべき次元より小さい次元にあるとき，その大きさは ∞ であると認識される．次の命題は，如何なる図形についても，0 以外の有限の大きさを取り得る次元は唯一であることを述べる．

命題 6.27. 任意の $A \subseteq \{0,1\}^{\mathbb{N}}$ について，次のような実数 $s \in [0,1]$ が存在する：

1. 任意の $t \in [0,s)$ について，$\lambda^t(A) = \infty$ である．
2. 任意の $t \in (s,1]$ について，$\lambda^t(A) = 0$ である．

定義 6.28 (ハウスドルフ次元 1917)**.** 集合 $A \subseteq \{0,1\}^{\mathbb{N}}$ の**ハウスドルフ次元** (*Hausdorff dimen-*

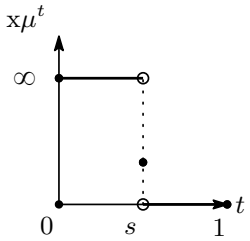


図 6　各集合には，高々 1 つの “ちょうどいい本来の次元” が存在し，それがハウスドルフ次元に他ならない．

sion) は次によって与えられる実数である．

$$\dim_H(A) = \inf\{s \in [0,1] : \lambda^s(A) = 0\}.$$

さて，いま $\alpha \in A \subseteq \{0,1\}^{\mathbb{N}}$ だということが分かっているとして，我々は α が何であるか知りたいとする．集合 A が小さければ小さいほど，α が何であるかの候補が絞られている．すなわち，候補集合 A が小さく特定されていればいるほど，α の値を予測しやすく，そして α の値当て賭博で儲けを得ることは容易となるであろう．さて，ハウスドルフ次元が 1 未満の集合は，単純に確率 0 であるというよりも，さらに小さい．Ville の定理 6.21 を思い出せば，確率 0 にまで候補を絞っていれば，好きなだけ儲けを出せるようなマルチンゲールを構成できた．それならば，ハウスドルフ次元 1 未満にまで候補を絞っていれば，更に儲けを出せると期待できる．

定義 6.29 (実効ハウスドルフ次元)**.** 次元として実数 $s \in [0,1]$ を固定する．集合 $A \subseteq \{0,1\}^{\mathbb{N}}$ が**実効的に λ^s-零**であるとは，ある $\{0,1\}^{<\mathbb{N}}$ の計算可枚挙集合列 $\{S_n\}_{n\in\mathbb{N}}$ で，任意の $n \in \mathbb{N}$ について次を満たすものが存在するときを指す：

$$A \subseteq [S_n], \text{ かつ } \sum_{\sigma\in S_n} 2^{-s|\sigma|} \leqslant 2^{-n}.$$

このとき，無限列 $\alpha \in \{0,1\}^{\mathbb{N}}$ の**実効ハウスドルフ次元** (*effective Hausdorff dimension*) は次によって与えられる実数である．

$$\dim_H(\alpha) = \inf\{s \in [0,1] : \{\alpha\} \text{ は実効的に } \lambda^s\text{-零である }\}.$$

まず，ハウスドルフ次元とマルチンゲールがどのように関連しているのかを確認しよう．

定理 6.30. 集合 $A \subseteq \{0,1\}^{\mathbb{N}}$ と実数 $s \in [0,1]$ について，次の 2 条件は同値である．

1. α の実効ハウスドルフ次元は s 以下である: $\dim_H(\alpha) \leqslant s$.
2. 次の条件を満たす下半計算可能マルチンゲール $d : \{0,1\}^{<\mathbb{N}} \to [0,\infty)$ が存在する：

$$\text{任意の } t > s \text{ について，} \quad \limsup_{n\to\infty} \frac{d(\alpha \upharpoonright n)}{2^{(1-t)n}} = \infty.$$

Proof. (1)⇒(2): $s > \dim_H(\alpha)$ を満たす実数 s を任意に取る．このとき，$\{\alpha\}$ は実効的 λ^s-零であるから，ある計算的枚挙可能集合列 $\{[U_n]\}_{n\in\mathbb{N}}$ が存在して，各 $n \in \mathbb{N}$ について，次の性質が満たされる．

$$\alpha \in [U_n], \text{ かつ } \sum_{\sigma\in U_n} 2^{-s|\sigma|} \leqslant 2^{-n}.$$

ここで，U_n は反鎖とすることができる．Ville の定理 6.21 のように，マルチンゲール d_n は U_n の条件付確率を模倣する．各 $\sigma \in \{0,1\}^*$ について，U_n^σ を，$\tau \sqsupseteq \sigma$ なる $\tau \in U_n$ 全体の集合とする．このとき，$d_n(\sigma)$ として，初期資金 $d_n(\langle\rangle) = \sum_{\sigma\in U_n} 2^{-s|\sigma|}$ であるような次の条件を満たすマルチンゲールを考える：

$$d_n(\sigma) = \begin{cases} 2^{|\sigma|}\sum_{\tau\in U_n^\sigma} 2^{-s|\tau|}, & \text{if } U_n^\sigma \neq \varnothing, \\ 2^{(1-s)m}, & \text{if } \sigma \upharpoonright m \in U_n \text{ for } m < |\sigma|, \\ 0, & \text{otherwise.} \end{cases}$$

このとき，$d(\sigma) = \sum_{n=1}^{\infty} d_n(\sigma)$ によって定義する．明らかに d は下半計算可能マルチンゲールである．いま $\alpha \in \bigcap_{n\in\mathbb{N}} U_n$ であるから，任意の $k \in \mathbb{N}$ について，$\alpha \upharpoonright n_k \in U_k$ なる $n_k \in \mathbb{N}$ が存在する．このとき，任意の実数 $t > s$ について，

$$\frac{d(\alpha\upharpoonright n_k)}{2^{(1-t)n_k}} \geqslant \frac{d_k(\alpha\upharpoonright n_k)}{2^{(1-t)n_k}} = \frac{2^{(1-s)n_k}}{2^{(1-t)n_k}} = 2^{(t-s)n_k}$$

であるから，次を得る．

$$\limsup_{n\to\infty} \frac{d(\alpha\upharpoonright n)}{2^{(1-t)n}} = \infty.$$

(2)⇒(1): いま，任意の実数 $t > s$ を固定する．各 $k \in \mathbb{N}$ について，$V_k \subseteq \{0,1\}^{<\mathbb{N}}$ を次によって定義する．

$$V_k = \left\{\sigma \in \{0,1\}^{<\mathbb{N}} : \frac{d(\sigma)}{2^{(1-t)|\sigma|}} \geqslant 2^k\right\}.$$

このとき，U_k を V_k の極小元のなす反鎖とする．つまり，U_k として，任意の $\tau \sqsubset \sigma$ について $\tau \in V_k$ であるような $\sigma \in V_k$ 全体の集合とする．このとき，次の不等式を得る．

$$\sum_{\sigma\in U_k} 2^{-t|\sigma|} \leqslant 2^{-k}\cdot\sum_{\sigma\in U_k} 2^{-t|\sigma|}\frac{d(\sigma)}{2^{(1-t)|\sigma|}} = 2^{-k}\cdot\sum_{\sigma\in U_k} 2^{-|\sigma|}d(\sigma) \leqslant 2^{-k}$$

.

ここで，最後の不等式は，定理 6.21 の証明中の主張による．主張 (2) の仮定より，$\alpha \in \bigcap_{n\in\mathbb{N}}[U_n]$ であるから，$\lambda^t(A)$ は実効的に零である．$t > s$ は任意なので，$\dim_H(\alpha) \leqslant s$ を得る． □

つづいて，ハウスドルフ次元とコルモゴロフ複雑性の関連性である．実は，ハウスドルフ次元の概念は，コルモゴロフ複雑性の極限的増大率，つまり「圧縮率」と密接に結び付いている．

定理 6.31. 無限列 $\alpha \in \{0,1\}^{\mathbb{N}}$ について，次の式が成立する．

$$\dim_H(\alpha) = \liminf_{n\to\infty} \frac{K(\alpha \upharpoonright n)}{n}.$$

Proof. まず，$\liminf_{n\to\infty} K(\alpha \upharpoonright n)/n < s$ を満たす有理数 $s \in \mathbb{Q}$ を任意に取る．このとき，任意の $k \in \mathbb{N}$ について，無限個の $n \in \mathbb{N}$ が存在して，$K(\alpha \upharpoonright n) \leqslant sn - k$ が成立する．任意の $k \in \mathbb{N}$ について，

$$U_k = \{\sigma \in \{0,1\}^{<\mathbb{N}} : K(\sigma) \leqslant s|\sigma| - k\}$$

と定義すれば，$\{U_k\}_{k\in\mathbb{N}}$ は計算可枚挙集合列である．このとき，次を得る．

$$\sum_{\sigma\in U_k} 2^{-(s|\sigma|-k)} \leqslant \sum_{\sigma\in U_k} 2^{-K(\sigma)} \leqslant \Omega \leqslant 1.$$

これより，$\sum_{\sigma\in U_k} 2^{-s|\sigma|} \leqslant 2^{-k}$ であり，$\alpha \in \bigcap_{k\in\mathbb{N}}[U_k]$ であるから，$\dim_H(\alpha) \leqslant s$ を得る．

逆に，$\dim_H(\alpha) < s$ なる有理数 $s \in \mathbb{Q}$ を取る．このとき，$\{0,1\}^{<\mathbb{N}}$ の計算可枚挙集合列 $\{U_n\}_{n\in\mathbb{N}}$ で，$\alpha \in \bigcap_n [U_n]$ かつ $\sum_{\sigma\in U_n} 2^{-s|\sigma|} \leqslant 2^{-n}$ を満たすものが存在する．もし，$\sigma \in \bigcup_{n\geqslant 1} U_n$ であることが分かったら，$(\sigma, s|\sigma|)$ を L に並べる．これはクラフトの不等式を満たす：

$$\sum_{(\sigma,r)\in L} 2^{-r} = \sum_{\sigma\in\bigcup_{n\geqslant 1} U_n} 2^{-s|\sigma|} \leqslant \sum_{n=1}^{\infty} \sum_{\sigma\in U_n} 2^{-s|\sigma|} \leqslant \sum_{n=1}^{\infty} 2^{-n} = 1.$$

よって，機械存在補題 6.16 より，ある定数 $c \in \mathbb{N}$ が存在して，各 $\sigma \in \bigcup_{n\geqslant 1} U_n$ について，$K(\sigma) \leqslant s|\sigma| + c$ を得る．無限個の $t \in \mathbb{N}$ が存在して，$\alpha \upharpoonright t \in \bigcup_{n\geqslant 1} U_n$ を満たすから，

$$\liminf_{n\to\infty} \frac{K(\alpha \upharpoonright n)}{n} \leqslant s$$

を得る．よって，定理は示された． □

6.6 ランダム列のハウスドルフ次元

歪んだコインから得られるランダム列 α は，当然ながら 0 と 1 の出現頻度が偏っており，公平なコイン投げの意味ではランダムではない．したがって，多くの有限部分 $\alpha \upharpoonright n$ をより短い列に圧縮できるだろう．では，具体的には，我々はこのランダム列 α をどれくらい圧縮できるだろうか？

定義 6.32. 実数 $p \in [0,1]$ の**シャノン・エントロピー** (*Shannon entropy*) とは，次によって定義される実数 $\mathcal{H}(p) \in [0,1]$ である．

$$\mathcal{H}(p) = -p\log p - (1-p)\log(1-p).$$

たとえば，$\mathcal{H}(1/2) = 1$ であり，$\mathcal{H}(0.7) < 0.3$ である．

定理 6.33. 任意の無限列 $\alpha \in \{0,1\}^{\mathbb{N}}$ および実数 $p \in [0,1]$ について，もし $\mathrm{Freq}(\alpha) = p$ ならば，α の実効ハウスドルフ次元は $\mathcal{H}(p)$ 以下である．つまり，以下の条件を満たす．

$$\liminf_{n\to\infty} \frac{K(\alpha \upharpoonright n)}{n} \leqslant \mathcal{H}(p).$$

Proof. $\sigma \in \{0,1\}^*$ と $i \in \{0,1\}$ が与えられたとき，$\#i(\sigma)$ で，$\sigma(n) = i$ なる $n \leqslant |\sigma|$ の数を表す．いま，$p = 1/2$ の場合は自明なので，$p > 1/2$ と仮定して一般性を失わない．$\delta \in (0, p - 1/2]$ を任意の有理数とする．$\mathrm{Freq}(\alpha) = p \geqslant 1/2 + \delta$ なので，次が成立する．

$$\limsup_{n\to\infty} \frac{\#0(\alpha \upharpoonright n)}{n} \geqslant \frac{1}{2} + \delta.$$

マルチンゲール d を，常に次の値が 0 であることに現在資金の 2δ 倍を賭ける戦略によって与える．つまり，

$$d(\alpha \upharpoonright n) = (1+2\delta)^{\#0(\alpha\upharpoonright n)} \cdot (1-2\delta)^{\#1(\alpha\upharpoonright n)}$$

とする．このとき，次の等式が導かれる．

$$\begin{aligned}\frac{\log d(\alpha \upharpoonright n)}{n} &= \frac{\#0(\alpha \upharpoonright n)}{n}\log(1+2\delta) + \frac{\#1(\alpha \upharpoonright n)}{n}\log(1-2\delta) \\ &= 1 + \frac{\#0(\alpha \upharpoonright n)}{n}\log\left(\frac{1}{2}+\delta\right) + \frac{\#1(\alpha \upharpoonright n)}{n}\log\left(\frac{1}{2}-\delta\right).\end{aligned}$$

いま，$p \geqslant 1/2 + \delta$ なので，これより，

$$\limsup_{n\to\infty} \frac{\log d(\alpha \upharpoonright n)}{n} \geqslant 1 + \left(\frac{1}{2}+\delta\right)\log\left(\frac{1}{2}+\delta\right) + \left(\frac{1}{2}-\delta\right)\log\left(\frac{1}{2}-\delta\right) = 1 - \mathcal{H}\left(\frac{1}{2}+\delta\right)$$

を得る．任意の正実数 $\varepsilon > 0$ に対して，$1/2 + \delta$ が十分 p に近いような δ を取れば，$\mathcal{H}(1/2+\delta) < \mathcal{H}(p) + \varepsilon$ が成立する．このような $\delta \in (0, p - 1/2]$ に対して，

$$\limsup_{n\to\infty} \left(\log d(\alpha \upharpoonright n) - n(1 - \mathcal{H}(p) - \varepsilon)\right) = \infty$$

であるから，次の式が成立する．

$$\limsup_{n\to\infty} \frac{d(\alpha \upharpoonright n)}{2^{n(1-\mathcal{H}(p)-\varepsilon)}} = \infty.$$

定数 $\varepsilon > 0$ は任意なので，定理 6.30 より，α の実効ハウスドルフ次元は $\mathcal{H}(p)$ 以下であることが示された． □

定理 6.34. $p \in (0,1)$ を任意の計算可能実数とし，λ_p をバイアス p のベルヌーイ測度とする．このとき，無限列 $\alpha \in \{0,1\}^{\mathbb{N}}$ が λ_p-ランダムならば，α の実効ハウスドルフ次元は，p のシャノン・エントロピー $\mathcal{H}(p)$ に等しい．つまり，以下の性質を満たす．

$$\liminf_{n\to\infty} \frac{K(\alpha \upharpoonright n)}{n} = \mathcal{H}(p).$$

Proof. α が λ_p-ランダムならば，Borel の強大数の法則を分析すると，$\mathrm{Freq}(\xi) \neq p$ なる $\xi \in \{0,1\}^{\mathbb{N}}$ 全体の集合は実効的に λ_p-零であることが分かる．これより，$\mathrm{Freq}(\alpha) = p$ であることが示される．よって，定理 6.33 より，$\dim_H(\alpha) \leqslant \mathcal{H}(p)$ が成立する．後は $\dim_H(\alpha) \geqslant \mathcal{H}(p)$ を示せばよい．任意の計算可能な正実数 $s < \mathcal{H}(p)$ を固定する．いま，マルチンゲール d を任意に取る．この λ-マルチンゲールと同じ方に同じ割合の額を賭けていく場合の，λ_p による資金過程 d_p を考える．つまり，λ_p-マルチンゲール d_p を，任意の $\sigma \in \{0,1\}^{<\mathbb{N}}$ について，次によって定義する：

$$d_p(\sigma) = \frac{2^{-|\sigma|}}{\lambda_p([\sigma])} d(\sigma).$$

いま，$\log \lambda_p([\sigma]) = \#0(\sigma)\log p + \#1(\sigma)\log(1-p)$ であることに注意する．ここで，$\#i(\sigma)$ によって $\sigma \in \{0,1\}^{<\mathbb{N}}$ 中に出現する $i \in \{0,1\}$ の総数を表す．もし $\mathrm{Freq}(\alpha) = p$ ならば，$\#0(\alpha \upharpoonright n)/n$ は p に収束するから，$\log \lambda_p([\alpha \upharpoonright n])/n$ は $-\mathcal{H}(p)$ に収束する．これより，十分大きな $n \in \mathbb{N}$ について，$sn + \log \lambda_p([\alpha \upharpoonright n]) < 0$ を得る．このような $n \in \mathbb{N}$ に対して，

$$d_p(\alpha \upharpoonright n) = \frac{2^{-n}}{\lambda_p([\alpha \upharpoonright n])} d(\alpha \upharpoonright n) = \frac{1}{2^{sn + \log \lambda_p([\alpha \upharpoonright n])}} \cdot \frac{d(\alpha \upharpoonright n)}{2^{(1-s)n}} > \frac{d(\alpha \upharpoonright n)}{2^{(1-s)n}}$$

が成立する．α は λ_p-ランダムなので，$\limsup_{n\to\infty} d_p(\alpha \upharpoonright n) < \infty$ であるから，右辺の上極限の値も有限に収束する．また，$s < \mathcal{H}(p)$ は任意なので，定理 6.30 より，$\dim_H(\alpha) \geqslant \mathcal{H}(p)$ を得る． □

おわりに

ここでは，各節を執筆する際に参考にした文献を紹介する．

■第 1 節「計算モデルと計算理論」 決定問題の節の証明を容易にするために，文字列書換系としてチューリング機械を導入した．しかし，記法があまり洗練されていないので，独習の際には若干読みにくいと感じるかもしれない．特に参考にした文献は存在しない．

■第 2 節「決定問題：解ける問題と解けない問題」 決定問題に関して，まとまったテキストが見つからなかったため，ウェブ上の講義ノート [9] を参考にした．決定問題についての様々な講義ノートが見つかるので，興味がある読者はウェブで検索して，色々と読み比べてみるのがよいと思う．決定問題について概観するならば，Poonen [6] が良い．

■第 3 節「部分組合せ代数」　チューリング還元などの話題は軽く流してしまっているので，もう少し知りたい場合は Cooper [3] か Soare [7] を参考にするとよい．部分組合せ代数の中でのラムダ計算の展開については，横内 [10] を参考にした．

■第 4 節「表現空間の理論」　表現空間の基本的な教科書は Weihrauch [8] であるが，表現空間の理論が本格的に進展したのが，Weihrauch [8] の出版以降なので，次世代の本格的な教科書の出版が待たれる．ここでの表現空間の解説は，Bauer [1] を参考にしている．また，本稿でのライスの定理の証明は，Bauer [2] の**総合計算可能性理論** (*synthetic computability theory*) を参考にした．

■第 5 節「極限計算可能性」　この節は，2017 年度秋期の講義では触れることができなかった．しかし，極限計算可能性と算術的階層について講義中で少しは触れるつもりで，講義ノートの該当部分は執筆中だった．このため，完成している部分のみ，ここで公開している．執筆には Downey-Hirschfeldt [5] の第 5 章を参考にしている．

■第 6 節「アルゴリズム情報理論」　何年か前の数学基礎論サマースクールの講義資料を改訂したものである．講義時間の制限のため，後半は駆け足になった．執筆には Downey-Hirschfeldt [5] の主に第 13 章を参考にした．

■計算可能性理論の教科書　講義時間の制限とシラバスによる制約もあり，計算可能性理論の極一部のトピックにしか触れることができなかった．特に，計算可能性理論の最も基礎的なトピックである算術的階層について全く触れられなかったのが心残りである．計算可能性理論の古典を概観したい読者には，Odifreddi [4] を勧める．証明なども含めて計算可能性理論の基礎をしっかり習得したい場合は，既に挙げた Cooper [3] か Soare [7] が良い．

計算可能性理論の各トピック毎に優れた教科書はいくつもあるが，紹介しているとキリが無いので，本稿ではここまでとしておく．

参考文献

[1] Andrej Bauer, First Steps in Synthetic Computability Theory, Electronic Notes in Theoretical Computer Science **155**, 12 (2006), pp. 5-31.

[2] Andrej Bauer, The realizability approach to computable analysis and topology. Ph.D. Thesis, Carnegie Mellon University. 2000. 248 pages.

[3] S. Barry Cooper, *Computability Theory*. Chapman & Hall/CRC, Boca Raton, FL, 2004. x+409 pp.

[4] Piergiorgio Odifreddi, *Classical Recursion Theory. The theory of functions and sets of natural numbers*. Studies in Logic and the Foundations of Mathematics, 125. North-Holland Publishing Co., Amsterdam, 1989. xviii+668 pp.

[5] Rodney G. Downey, Denis R. Hirschfeldt, *Algorithmic Randomness and Complexity*. The-

ory and Applications of Computability. Springer, New York, 2010. xxviii+855 pp.

[6] Bjorn Poonen, Undecidable problems: a sampler, available at https://arxiv.org/abs/1204.0299

[7] Robert I. Soare, *Turing Computability. Theory and Applications*, Theory and Applications of Computability. Springer-Verlag, Berlin, 2016. xxxvi+263 pp.

[8] Klaus Weihrauch, *Computable Analysis. An Introduction*. Texts in Theoretical Computer Science. An EATCS Series. Springer-Verlag, Berlin, 2000. x+285 pp.

[9] Michael M. Wolf, Lecture on undecidability, available at https://www-m5.ma.tum.de/foswiki/pub/M5/Allgemeines/MA5116_2012S/lecture.pdf

[10] 横内寛文『プログラム意味論』共立出版, 1994 年.