42. あつかいにくい問題†

小林　孝次郎††

1. はじめに

与えられた入力に対しある出力を出す，あるいは，yes, no の答を出す，という種類の問題を考えるとき，ある問題はやさしく，ある問題は難しく感じられる．我々がそう感じるのは直観的な意味においてであるが，最近の計算量理論の発展により，ある種の問題が**本質的に難しい問題(あつかいにくい問題**, intractable problem) であることを，厳密に証明することが可能になった．さらに，現実に我々が興味を持つ具体的な問題の中にもこのような問題が存在することがわかってきた．本稿ではそのような具体的な問題の1つを示し，その困難さをどのようにして「証明する」ことができるのか，の概略を述べる．

なおここで本質的に難しい問題（あつかいにくい問題）というのは，「原理的にはアルゴリズムによって解けるが，実際に解く作業が大変であるような問題」であって，解くアルゴリズムそのものが存在しない，いわゆる決定不能な問題 (undecidable problem) ではないことを，おことわりしておく．

2.「本質的に難しい問題」とは

ある問題が「本質的に難しい問題である」ということは，その問題を解くどのようなアルゴリズムも効率が悪い，ということである．そこでこの概念を厳密に定義するためには，(1)アルゴリズムとしてどのようなものを考えるか，(2)アルゴリズムの効率をどのように測るか，(3)「効率が悪い」とはどの程度悪いことを指すか，という3つのことをはっきりさせなければならない．

我々はアルゴリズムとして Turing 機械（以後 Tm と略す）を採用する．一応 Tm の基礎概念については既知とするが（文献 1)～3) 等を参照されたい），我々が採用する Tm のモデルについて少し説明する．

我々の Tm M は，**図-1** に示すような，右方向に無限にのびる1本のテープを持つ決定性 Tm である．M への入力は，入力アルファベットと呼ばれるあるアルファベット Σ の語 x である．入力 x は，テープの初期の内容を xBB... にしておくことによって M に与えられる (B ($\notin\Sigma$) は空白記号)．テープには $\Sigma\cup\{B\}$ 以外の記号も記入することができる．テープに記入できるすべての記号の集合 Γ を，テープアルファベットと呼ぶ ($\Sigma\cup\{B\}\subseteq\Gamma$)．

L を Σ^* のある部分集合とする．(Σ^* は，アルファベット Σ のすべての語の集合を表わす．) x ($\in\Sigma^*$) を M に与えると，$x\in L$ なら受理して止まり $x\notin L$ なら受理しないで止まるとき，M は L を決定するという．

アルゴリズムの効率としては，我々は Tm の計算時間，あるいは使用テープ領域量を採用する．長さ n の入力を M に与えると $f(n)$ ステップ以内に必ず止まるとき，M は時間的な複雑さが $f(n)$ の Tm であるという．同様に，長さ n の入力を M に与えると，ヘッドが左から $f(n)$ 番目のマス目より右にはみ出ないような動きをして止まるとき，M は空間的な複雑さが $f(n)$ の Tm であるという．時間的な複雑さ（あるいは空間的な複雑さ）が $f(n)$ であるような Tm によって決定される集合のクラスを DTime[$f(n)$]（あるいは DSpace [$f(n)$]）で表わすことにする．(文字 D は，決定性 (deterministic) Tm を使っていることを意味する．)

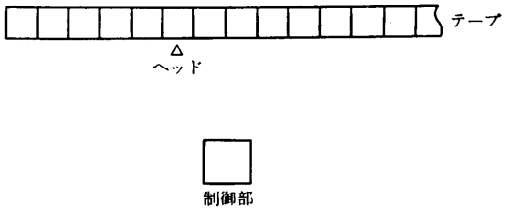


図-1 Turing 機械

† Intractable Problems by Kojiro KOBAYASHI (Department of Information Science, Tokyo Institute of Technology).

†† 東京工業大学理学部情報科学科

「効率が悪い」という概念については，ここでは一応「計算時間が入力の長さの多項式ではおさえられない」ということを定義として採用する．

以上の考え方にもとづき我々は，どのような多項式 $p(n)$ に対しても $L \notin \mathrm{DTime}[p(n)]$ であるとき，L の決定問題（与えられた語 x に対し $x \in L$ か否かを決定する問題）は「本質的に難しい」という，と約束する．

3. 平方演算を許す正規表現

$\Delta = \{a_1, \cdots, a_m\}$ をあるアルファベットとする．このとき，**正規表現**というものを次の規則によって定義する．

（1） $\phi, a_1, \cdots, a_m$ は正規表現である．

（2） α が正規表現なら，α^*, α^2 もそうである．

（3） α, β が正規表現なら，$(\alpha \cdot \beta), (\alpha \cup \beta)$ もそうである．

この正規表現の定義は，α^2 を許す点でふつうの定義とちがっている．区別する必要がある場合には，我々の定義による正規表現を「**平方演算を許す正規表現**」と呼ぶことにする．さらに Δ を明示したい場合には，「Δ 上の（平方演算を許す）正規表現」という．

Δ 上の正規表現 α に対し，次のような規則で Δ^* の部分集合 $L(\alpha)$ を対応づける．

$L(\phi) = \{\ \}$（空集合），$L(a_i) = \{a_i\}$，

$L(\alpha^*) = \{u_1 \cdots u_n \mid n \geq 0$，各 u_i は $L(\alpha)$ に含まれる$\}$，

$L(\alpha^2) = \{u_1 u_2 \mid u_1, u_2$ は $L(\alpha)$ に含まれる$\}$，

$L((\alpha \cdot \beta)) = \{u_1 u_2 \mid u_1$ は $L(\alpha)$ に，u_2 は $L(\beta)$ に含まれる$\}$，

$L((\alpha \cup \beta)) = \{u \mid u$ は $L(\alpha)$ または $L(\beta)$ に含まれる$\}$．

$L(\alpha)$ を，正規表現 α の表わす集合と呼ぶ．

$\Delta = \{0, 1\}$ の場合の正規表現のいくつかについて，それらの表わす集合を示す．

$L((0 \cup 1)^*)$ = すべての語の集合（$= \Delta^*$），

$L(((0 \cup 1)^* \cdot 0))$ = 0 で終わるすべての語の集合，

$L((0 \cup 1)^2) = \{00, 01, 10, 11\}$，

$L((0 \cup 1)^{2^*})$ = 長さが偶数のすべての語の集合，

$L(1^{222}) = \{11111111\}$，

$L(\phi^*) = \{\epsilon\}$（ϵ は空の語）．

Δ 上の正規表現そのものも，$\Delta' = \Delta \cup \{\phi, *, ^2, \cdot, \cup,), (\}$ というアルファベットの語である．そこで語の集合 $L_{\Delta,\mathrm{reg}}$ $(\subseteq \Delta'^*)$ を

$L_{\Delta,\mathrm{reg}} = \{\alpha \mid \alpha \in \Delta'^*$，$\alpha$ は Δ 上の平方演算を許す正規表現，$L(\alpha) \neq \Delta^*\}$

と定義する．

正規表現と有限オートマトンの関係を利用することにより，$L_{\Delta,\mathrm{reg}}$ を決定する Tm が存在することを，容易に示すことができる．しかし我々は，$L_{\Delta,\mathrm{reg}}$ の決定問題が2節で定義したような意味で，本質的に難しい問題であることを証明することができる．

4.「難しさ」の証明

証明は，2つの補助定理に分けると理解しやすい．

補助定理 1. DSpace $[2^{2^n}]$ に含まれ DSpace $[2^n]$ に含まれない集合 L_0 が存在する．

（証明） 対角線論法と呼ばれる方法による．

Σ を2つの記号 0, 1 よりなるアルファベットとする．Σ を入力アルファベットとして持つ任意の Tm M の構造を，Σ のある語 $\mathrm{des}(M)$ で表わす方法を決めておく．$\mathrm{des}(M)$ は，M が何個の状態を持つか，どの状態が初期状態でどの状態が受理状態か，M のテープアルファベットは何個の記号を含むか，どのような状態でどういう記号を見たとき，M はどのような動作をするか等を，0, 1 だけを使って表現したものである．

Σ を入力アルファベットとして持つある Tm M_0 を定義する．長さ n の入力 x $(\in \Sigma^*)$ を M_0 に与えたとき，M_0 は次のような動作をおこなう．

（1） テープ上で，左から 2^{2^n} 番目のマス目に，ある記号を書く．ただしそのための動作のあいだ，ヘッドが 2^{2^n} 番目のマス目より右にはみ出ることのないようにする．次に(2)に進むが，(2)，(3)の動作中にヘッドが 2^{2^n} 番目のマス目より右にはみ出ようとすると，M_0 は入力を受理しないで止まってしまうものとする．

（2） x が 00…01 $\mathrm{des}(M)$ というかたちをしているか否かを調べる．no なら入力を受理しないで止まる．yes なら(3)に進む．

（3） $\mathrm{des}(M)$ の部分を解読しながら，M に x $(=$ 00…01 $\mathrm{des}(M))$ を与えたときの動きをシミュレートする．M が x を受理して止まれば M_0 は x を受理しないで止まり，M が x を受理しないで止まれば M_0 は x を受理して止まる．M がいつまでも止まらない場合には，M_0 はそのことを検出することができる．その場合には，M_0 は x を受理して止まる．

M_0 が M のシミュレーションをおこなうとき，M

の状態やテープ上の記号は，M_0 のテープの上で M_0 の記号の列として符号化して表現される．したがって，p 個のマス目を使う M の動きを，M_0 は $c_M p + c_M'$ 個のマス目を使ってシミュレートすることができる．ここで c_M, c_M' は，M によって決まる定数である．

M_0 はどんな入力に対しても止まる．M_0 が決定する言語を $L_0 (\subseteq \Sigma^*)$ とする．M_0 は 2^{2^n} 個のマス目しか使わないから，L_0 は DSpace $[2^{2^n}]$ に含まれる．この L_0 が，空間的な複雑さが 2^n のある Tm M_1 で決定できるとして矛盾を導く．

i を

$$2^{|0^i 1 \mathrm{des}(M_1)|} c_{M_1} + c_{M_1}' \leq 2^{2^{|0^i 1 \mathrm{des}(M_1)|}}$$

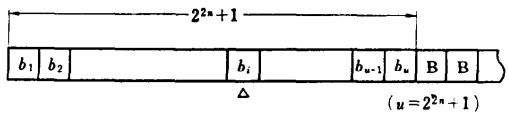
がなりたつようなある値とし，$x = 0^i 1 \mathrm{des}(M_1)$ を M_0 に与えた場合の M_0 の動きがどうなるかを考えてみる．x の長さを n とする．

M_0 の動きはステップ(3)に進み，M_1 に x を与えた場合のシミュレーションをおこなう．M_1 はたかだか 2^n 個のマス目しか使わないから，そのシミュレーションにおいて，M_0 はたかだか $2^n c_{M_1} + c_{M_1}' \leq 2^{2^n}$ 個のマス目しか使わない．したがって，ステップ(1)の最後に示した条件によってシミュレーションが中断されることはない．そこで我々は，次のような矛盾を得る．

$x \in L_0$
$\Leftrightarrow M_1$ は x を 受理して止まる
　(M_1 は L_0 を決定するから)
$\Leftrightarrow M_0$ は x を受理しないで止まる
　(M_0 のステップ(3)の決め方により)
$\Leftrightarrow x \notin L_0$
　(M_0 も L_0 を決定するから)．　(証明終り)

補助定理 2. $L_0 (\subseteq \Sigma^*)$ が DSpace $[2^{2^n}]$ に含まれるなら，任意の $x (\in \Sigma^*)$ に対し Δ 上の平方演算を許す正規表現 α_x を対応づける対応で，次の3つの条件を満足するものが存在する：(1) $x \in L_0 \Leftrightarrow L(\alpha_x) \neq \Delta^*$，(2) $|\alpha_x| \leq d|x| + d'$，(3) x から α_x を作り出す計算は，空間的な複雑さが $dn + d'$ の Tm で実行できる $(n = |x|)$．ここで，Δ は L_0 のみによって決まるアルファベット，d, d' は L_0 のみによって決まる定数である．

(証明)　M_0 を，L_0 を 決定する空間的な複雑さが 2^{2^n} の Tm とし，S を M_0 の状態の集合，Γ を M_0 のテープアルファベットとする $(\Sigma \cup \{B\} \subseteq \Gamma)$．アルファベット Δ を $\Delta = \Gamma \cup S \times \Gamma \cup \{\#\}$ で定義する．



制御部の状態は s

図-2　Turing 機械の状況

$x = a_1 \cdots a_n$ $(\in \Sigma^*)$ を長さ n の語とし，x を入力として与えたときの M_0 の動きを考える．

この動きの途中において，M_0 の状況は常に**図-2**のようなかたちになっている．この状況を，Δ の語 $\delta = b_1 \cdots b_{i-1}(s, b_i) b_{i+1} \cdots b_{2^{2^n}+1}$ で表わすことにする．$(b_1, \cdots, b_{i-1}, b_{i+1}, \ldots, b_{2^{2^n}+1}$ は Γ の要素，(s, b_i) は $S \times \Gamma$ の要素である．) $\delta_1, \cdots, \delta_t$ が M_0 の状況を表わす語の列であるとき，この列を再び Δ の1つの語 $\#\delta_1\#\delta_2\#\cdots\#\delta_t\#$ で表わすことにする．

$x \in L_0$，つまり M_0 が x を受理して止まるということと，次の条件を満足する $\#\delta_1\#\delta_2\#\cdots\#\delta_t\#$ 型の語が存在する，ということは同値である．

(a1)　$\delta_1 = (s_1, a_1) a_2 \cdots a_n$ BB$\cdots$B　(s_1 は初期状態，B は $2^{2^n}+1-n$ 個)．

(a2)　各 $i\ (1 \leq i < t)$ に対し，δ_{i+1} は δ_i が表わす M_0 の状況の1ステップ後の状況を表わす．

(a3)　δ_t は $F \times \Gamma$ の要素を含む．ただし F は受理状態の集合である．

そこで，上の条件を満足する $\#\delta_1\#\delta_2\#\cdots\#\delta_t\#$ 型の語ではないような Δ の語の集合を表わす正規表現がもしあれば，それを d_x とすれば補助定理の条件(1)がなりたつ．

上の条件(a1)～(a3)を満足する $\#\delta_1\#\delta_2\#\cdots\#\delta_t\#$ 型の語ではないという条件は，次のような7つの条件(b1)～(b7)の論理和である．

(b1)　$\#a_1 a_2 \cdots a_n$ ではじまらない．

(b2)　# で終わらない．

(b3)　# でない 2^{2^n} 個以下の記号をはさむ2つの # が存在する．

(b4)　# でない $2^{2^n}+2$ 個以上の記号をはさむ2つの # が存在する．

(b5)　$\#a_1 a_2 \cdots a_n$ と第2の # の間に，B でない記号がある．

(b6)　条件(a2)がなりたたない．

(b7)　条件(a3)がなりたたない．

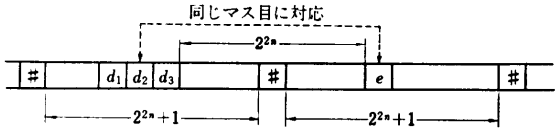


図-3 Turing 機械の動きの記述

これらの条件のそれぞれに対し，それを満足する語の集合を表わす正規表現があることを示す．

まず (b1) は

$$((\Delta-\{\#\})\cup\#((\Delta-\{a_1\})\cup a_1((\Delta-\{a_2\})\cup a_2(\cdots\cup((\Delta-\{a_{n-1}\})\cup a_{n-1}(\Delta-\{a_n\}))\cdots))))\Delta^*$$

で表わせる．ただし 例えば $(\Delta-\{\#\})$ は，$\Delta-\{\#\}$ に含まれるすべての記号を∪でつないだ正規表現を意味する．(b3) は

$$\Delta^*\#(\Delta\cup\phi^*)^{2^{2n}-2}\#\Delta^*$$

で表わせる．(^{2} は $2n$ 個．$L(\phi^*)=\{\epsilon\}$ であることに注意．) (b2)，(b4)，(b5)，(b7)も容易に正規表現で表わせる．最後に (b6) 条件について考える．

図-3 に示すように，d_2 と e が同じマス目に対応する位置の記号であり，d_1, d_3 が d_2 の両側の記号であるとする．条件 (a2) がなりたつとき，e は $d_1d_2d_3$ からただ1つに決まる．そこで条件 (b6) は，上記の決め方に違反するような組み合わせの $d_1d_2d_3$ と e が，長さ 2^{2n} をへだてて現われる，ということである．そこで，違反するようなすべての $d_1d_2d_3$ と e の組み合わせに対する

$$\Delta^*d_1d_2d_3\Delta^{2^{2n}-2}e\Delta^*$$

(^{2} の数は $2n$ 個) を∪でつないだ正規表現が，条件 (b6) を表わす．

このようにして得られた，条件 (b1)～(b7) を表わす7つの正規表現を∪でつないだものを α_x とすればよい．この α_x が補助定理の残りの条件(2)，(3)をも満足するように，α_x を作ることが可能であることは，本質的な困難さなしに証明することができる．

(証明終り)

いよいよ主要結果を示すことができる．

定理 $L_{\Delta,reg}\notin$DSpace $[2^{cn}]$ であるようなアルファベット Δ と定数 c (>0) が存在する．

(証明) どんな Δ と c (>0) に対しても $L_{\Delta,reg}\in$ DSpace $[2^{cn}]$ がなりたつと仮定して矛盾を導く．

補助定理1により，$L_0\subseteq\Sigma^*$，$L_0\in$DSpace $[2^{2n}]-$DSpace $[2^n]$ であるようなアルファベット Σ と集合 L_0 が存在する．この Σ と L_0 に対し，補助定理2の条件(1)～(3)がなりたつようなアルファベット Δ，定数 d, d' と x から α_x への対応を考える．

仮定により，$L_{\Delta,reg}$ を決定する空間的な複雑さが $2^{n/2d}$ の Tm M_2 が存在する．そこで，入力 x $(\in\Sigma^*)$ を α_x に変換し，それを M_2 に与えるような Tm を M_3 とすれば，

$$\begin{aligned}x\in L_0&\Leftrightarrow L(\alpha_x)\neq\Delta^*\\&\Leftrightarrow\alpha_x\in L_{\Delta,reg}\\&\Leftrightarrow M_2\text{ は }\alpha_x\text{ を受理して止まる}\\&\Leftrightarrow M_3\text{ は }x\text{ を受理して止まる}\end{aligned}$$

がなりたつから，M_3 は L_0 を決定する．しかも x の長さ n が十分大きい場合には，M_3 の使うマス目の数は

$$\max\{n+1, dn+d', 2^{(dn+d')/2d}\}\leq 2^n$$

以下である．また n が小さい場合には，M_3 が $n+1$ $(\leq 2^n)$ 個のマス目しか使わないようにできる．したがって M_3 は空間的な複雑さが 2^n の Tm であると考えてもよく，$L_0\in$DSpace $[2^n]$ がなりたつ．これは矛盾である． (証明終り)

系 どのような多項式 $p(n)$ に対しても $L_{\Delta,reg}\notin$ DTime $[p(n)]$ がなりたつような，アルファベット Δ が存在する．

(証明) DTime $[p(n)]\subseteq$DSpace $[\max\{n+1, p(n)+1\}]$[1] と上の定理より明らか． (証明終り)

これで $L_{\Delta,reg}$ の決定問題が，本質的に難しい問題であることが示せた．

5. 結 論

原理的には解くことができるが，実際的には「解けない」とみなしてよい (そしてそのことを厳密に証明できる)ような問題の実例を，1つ示した．現在のところ，オートマトン理論や論理学の分野においてこのような問題が知られているが[4)~6)]，今後他の色々な分野においてもこのような問題が見い出されるであろう．

この種の問題を解くことが実際に必要になった場合には，(1)問題を特殊な場合のみにしぼって効率のよいアルゴリズムを考察する，(2)確率アルゴリズムや，場合によっては誤った答を出すアルゴリズム等，アルゴリズムの概念を変える，といったことが必要であろう．

参考文献

1) Hopcroft, J.E. and Ullman, J.D.: Introduction to Automata Theory, Languages, and Computation, p. 418, Addison-Wesley Publishing Co., Reading (1979).
2) 福村晃夫, 稲垣康善: オートマトン・形式言語理論と計算論, p. 235, 岩波書店 (1982).
3) Garey, M.R. and Johnson, D.S.: Computers and Intractability, A Guide to the Theory of NP-Completeness, p. 338, W.H. Freeman and Co., San Francisco (1979).
4) Meyer, A.R. and Stockmeyer, L.J.: The Equivalence Problem for Regular Expressions with Squaring Requires Exponential Space, Proc. 13 IEEE Symp. on Switching and Automata Theory, pp. 125-129 (1973).
5) Fischer, M.J. and Rabin, M.O.: Super-exponential Complexity of Presburger Arithmetic, Proc. AMS Symp. on Complexity of Real Computational Processes, Vol. VII (1974).
6) Ferrante, J. and Rackoff, C.W.: The Computational Complexity of Logical Theories, Lecture Notes in Math., Vol. 718, Springer-Verlag, Berlin, Heidelberg, New York (1979).

(昭和57年12月6日受付)