■6 群（コンピュータ‐基礎理論とハードウェア）- 2 編（計算論とオートマトン）

5 章 決定性，非決定性計算の複雑さ

（執筆者：河内亮周）［2010 年 4 月 受領］

■概要■

決定性，非決定性計算はアルゴリズム理論，計算量理論を考える上で最も基本かつ重要な計算モデルである．決定性計算は与えられた入力に対して計算結果つまり出力が一意に決定される計算である．通常のコンピュータ上で動作するアルゴリズム及びコンピュータが解くべき計算問題は，多くの場合この計算モデル上で理論的解析がなされている．一方，非決定性計算は問題の「答え」が与えられた場合に，その答えの正しさを検証するための決定性計算をモデル化している．多くの実用的な計算問題が非決定性計算で効率良く解けることが知られているが，決定性計算で効率良く解けるのか分かっていない．これが理論計算機科学分野で最も有名な未解決問題「NP ≠ P 予想」である．この未解決問題が注目されるようにこれらの計算モデル上での計算複雑さを理論的に明らかにすることは計算機科学の基礎として非常に重要である．

【本章の構成】

まず，効率の良い決定性，非決定性計算を特徴付ける計算量クラスとしてクラス P と NP の解説を（5-1）節で行う．また，決定性，非決定性計算の計算時間に関する計算能力の差異を示した階層定理を（5-2）節で解説する．（5-3）節では非決定性計算をさらに一般化した概念である交代性計算を解説する．また，計算時間以外の計算複雑さを測る指標として使用するメモリの大きさ，つまり，領域複雑さに関する解説を（5-4）節で行う．

■6 群 - 2 編 - 5 章

5-1 クラス P と NP

（執筆者：荻原光徳）[2009 年 1 月 受領]

5-1-1 P=?NP 問題と NP 完全性

チューリング機械 M が，すべての入力 x に対して x の長さのある多項式時間以内に停止するとき，M は多項式時間限定であるという．P と NP はそれぞれ，多項式時間限定のチューリング機械によって判定可能な集合全体のクラス，多項式時間限定の非決定性チューリング機械によって判定可能な集合全体のクラスである．定義より $\mathrm{P} \subseteq \mathrm{NP}$ は自明だが，$\mathrm{P} = \mathrm{NP}$ か否かは判明していない．これを P=?NP 問題と呼ぶ．

P=?NP 問題に関係する重要な概念の一つは，NP 完全性[6, 9, 12]である，集合 A, B に対して，多項式時間計算可能関数 f で，条件 $(\forall x)[x \in A \Leftrightarrow f(x) \in B]$ を満たすものが存在するとき，A は B へ多項式時間多対一還元可能であるといい，$A \leq_m^p B$ と表記する．NP の集合 C が $(\forall A \in \mathrm{NP})[A \leq_m^p C]$ という条件を満たすとき，C は NP 完全であるといい，NP 完全集合のクラスを NPC で表す．本質的に，NPC は NP の中で最も難しい集合のクラスである．何万にものぼる実用的かつ NP 完全な判定問題が発見されており[8]，代表としてグラフ彩色問題（どの隣り合う頂点も同じ色をもたないように，グラフの頂点を k 色を用いて塗り分けることができるか否か判定する問題）や充足可能性問題（命題論理式の値が真になるような論理変数の割当てがあるか否か判定する問題）があげられる．定義より $\mathrm{NPC} \subseteq \mathrm{P}$ または $\mathrm{NPC} \cap \mathrm{P} = \emptyset$ である．また，$\mathrm{P} \neq \mathrm{NP}$ ならば，$\mathrm{NP} - (\mathrm{NPC} \cup \mathrm{P}) \neq \emptyset$ であることが判明している[11]．

5-1-2 同型仮説と疎集合

集合 A, B に対して，全単射 f で，$\{f(x) \mid x \in A\} = B$，$\{f(x) \mid x \in \overline{A}\} = \overline{B}$，かつ f と f^{-1} がどちらも多項式時間計算可能，という条件を満たすものが存在するとき，A と B は多項式時間同型であるといい，$A \equiv_{\mathrm{iso}}^p B$ と表記する．同型仮説[5]とは，$(\forall A, B \in \mathrm{NPC})[A \equiv_{\mathrm{iso}}^p B]$，というもので，あらゆる NP 完全集合が，単一の集合の並び替えとして構成できるということを意味する．$\mathrm{P} = \mathrm{NP}$ であれば，NP の集合は，空集合と全体集合以外はすべて NP 完全である．このことから，同型仮説が成り立つならば $\mathrm{P} \neq \mathrm{NP}$ であることが容易に導ける．しかし，逆に $\mathrm{P} \neq \mathrm{NP}$ の場合に同型仮説が成り立つかどうかは判明していない．

集合 A の濃度とは，各自然数 n に，長さ n 以下の A の要素数を対応させる関数である．濃度が多項式以下であるような集合を疎集合と呼ぶ．同型仮説が成り立つとすると，NP 完全集合はすべて指数関数程度の濃度をもち、よって疎な NP 完全集合は存在しない．実際，疎な NP 完全集合が存在するとき，またそのときに限り $\mathrm{P} = \mathrm{NP}$ であることが知られている[13]．この結果を強めて，多項式時間多対一還元可能性の拡張の一つである，多項式時間因数限定還元可能性のもとで NP 完全な疎集合が存在すれば $\mathrm{P} = \mathrm{NP}$ であることが判明している[14]．

5-1-3 相対化

チューリング機械に，集合 R の判定問題を単位時間で解く仮想的なアルゴリズムを与えることを，オラクル R による相対化という．計算量理論における未解決問題に対して，それを肯定するオラクルと，否定するオラクルが見つかれば，その問題は，計算モデルの模倣のような，相対化が可能な手法では解くことができない．P=?NP 問題に関しては，$\mathrm{P} = \mathrm{NP}$ を成

立させるオラクルと P ≠ NP を成立させるオラクルが発見されている[3]．前者は同型仮説を成立させないオラクルでもある．また，同型仮説を成立させるオラクルも発見されている[7]．よって，P=?NP 問題や同型仮説を解決するには，相対化できない手法が必要である．

オラクルを一様な確率分布のもとで選んだ場合，条件 Q が成立する確率は 0 または 1 である．そこで，相対化において成立する確率が 1 であることを，その条件が成り立つことの証拠として捉えようという主張がある．P ≠ NP の確率は 1 であり[4]，同型仮説の確率は 0 であることが知られている[10]．

5-1-4 証明の検定と PCP 定理

NP の集合とは，集合に属することの証明が，入力の長さのある多項式の長さをもち，決定性多項式時間で検定できるもの，といえる．最近これが，指数関数的に長い証明の数ビットのみを確率的に選んで検定できるもの，と定義できることが判明し[1]，その結果を PCP 定理と呼ぶ．多くの NP 完全集合には，ごく自然な計算問題が付随するが（例えばグラフ彩色問題において，彩色に必要な最小の色の個数を求める問題），PCP 定理を用いて，そのような計算問題の近似解を求める難しさの研究に著しい進歩が見られている[2]．

■参考文献

1) S. Arora and S. Safra, “Probabilistic checking of proofs: a new characterization of NP,” J. Assoc. Comput. Mach., vol.45, no.1, pp.70-122, 1998.
2) S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy, “Proof Verification and the Hardness of Approximation Problems,” J. Assoc. Comput. Mach., vol.45, no.3, pp.501-555, 1998.
3) T.P. Baker, J. Gill, and R. Solovay, “Relativizations of the P=?NP question,” SIAM J. Comput., vol.4, no.4, pp.431-442, 1975.
4) C.H. Bennet and J. Gill, “Relative to a random oracle, $P^A \neq NP^A \neq co\text{-}NP^A$ with probability 1,” SIAM J. Comput., vol.10, no.1, pp.96-113, 1981.
5) L. Berman and J. Hartmanis, “On isomorphisms and density of NP and other complete sets,” SIAM J. Comput., vol.6, no.2, pp.305-322, 1977.
6) S.A. Cook, “The complexity of theorem-proving procedure,” in Proceedings of the 3rd Annual ACM Symposium on the Theory of Computing, pp.151-158, ACM Press, Ney York, 1971.
7) S. Fenner, L.J. Fortnow, and S.A. Kurtz, “The Isomorphism Conjecture holds relative to an oracle,” SIAM J. Comput., vol.25, no.1, pp.193-206, 1996.
8) M.R. Garey and D.S. Johnson, “Computer and Intractability: A Guide to the Theory of NP-Completeness,” W.H. Freeman, New York, 1979.
9) R.M. Karp, “Reducibility among combinatorial problems,” in Complexity of Computer Computations, ed. by R. Miller and J. Thatcher, pp.85-104, Plenum Press, New York, 1972.
10) S.A. Kurtz, S.R. Mahaney, and J.S. Royer, “The isomorphism conjecture fails relative to a random oracle,” J. Assoc. Comput. Mach., vol.42, no.2, pp.401-420, 1995.
11) R.E. Ladner, “On the structure of polynomial time reducibility,” J. Assoc. Comput. Mach., vol.22, no.1, pp.155-171, 1975.
12) L.A. Levin, “Universal sequential search problems,” Prob. Inform. Trans., vol.9, no.3, pp.265-266, 1973.
13) S.R. Mahaney, “Sparse complete sets for NP: solution of a conjecture of Berman and Hartmanis,” J. Comput. Syst. Sci., vol.25, no.2, pp.130-143, 1982.
14) M. Ogiwara and O. Watanabe, “On polynomial time bounded truth-table reducibility of NP sets to sparse sets,” SIAM J. Comput., vol.20, no.3, pp.471-483, 1991.

■6 群 - 2 編 - 5 章

5-2 階層定理

（執筆者：岩本宙造）［2008 年 12 月 受領］

計算時間や記憶領域といった計算資源を，より多く用いれば，より難しい関数の計算や，より多くの言語の受理が可能になると考えられる．この性質を理論的に証明したものが，チューリング機械（Turing Machine: TM）の階層定理である．本節では，時間計算量と領域計算量に関する決定性・非決定性クラスの階層定理と証明手法について解説する．

5-2-1 決定性計算量の階層性と対角線論法

$s(n)$ 領域（$t(n)$ 時間）の決定性 TM で受理できる言語のクラスを，DSPACE($s(n)$) (DTIME($t(n)$)) と表す．このとき，$\inf_{n\to\infty} \frac{s_1(n)}{s_2(n)} = 0$ を満たす任意の領域構成可能関数 $s_1(n), s_2(n) \geq \log n$ に対して*, DSPACE($s_1(n)$) $\subsetneq$ DSPACE($s_2(n)$) という領域階層定理が成立する[2]．これは，言語受理の複雑さを階級づける尺度として，記憶領域の量が妥当なものであるという理論的な裏付けになっている．一方，任意の定数 $c > 0$ に対し，線形圧縮定理 DSPACE($s(n)$) = DSPACE($cs(n)$) が成立する[2]．この定理から，定数係数は計算複雑さの尺度にはならないことが分かる．非決定性の領域量（決定性・非決定性の時間量）でも同様の線形圧縮（加速）定理が知られている[3]．時間量に関しては，領域量ほど稠密ではないが，$\inf_{n\to\infty} \frac{t_1(n)\log t_1(n)}{t_2(n)} = 0$ なる任意の時間構成可能関数 $t_1(n), t_2(n)$ に対して，時間階層定理 DTIME($t_1(n)$) $\subsetneq$ DTIME($t_2(n)$) が成り立つ[3]．

計算量の代表的なクラスとして，L = DSPACE($\log n$), P = $\bigcup_{i\geq 1}$ DTIME(n^i), PSPACE = $\bigcup_{i\geq 1}$ DSPACE(n^i), EXP = $\bigcup_{i\geq 1}$ DTIME(2^{n^i}), EXPSPACE = $\bigcup_{i\geq 1}$ DSPACE(2^{n^i}) があり，更に，それらの非決定性に対応するクラスとして，NL, NP, NPSPACE, NEXP, NEXPSPACE がある．Savitch の定理 NSPACE($s(n)$) = DSPACE($s(n)^2$) から，これらには，L $\subseteq$ NL $\subseteq$ P $\subseteq$ NP $\subseteq$ PSPACE = NPSPACE $\subseteq$ EXP $\subseteq$ NEXP $\subseteq$ EXPSPACE = NEXPSPACE なる関係がある．これら関係のなかで，真の包含性が証明できているのは，上記の階層定理から導かれる NL $\subsetneq$ PSPACE $\subsetneq$ EXPSPACE と P $\subsetneq$ EXP だけである．

決定性の計算量クラスの階層定理の証明は，対角線論法による．任意の TM は，アルファベット $\{0,1\}$ 上の語として符号化できる．そこで，符号化列が e の TM を M_e と記し，言語 L を $L = \{ e2^i \mid \text{TM } M_e \text{ は，入力列 } e2^i \text{ を } s_1(|e2^i|) \text{ 領域以内では受理しない} \}$ と定義する．すると，L は，いかなる $s_1(n)$ 領域の TM でも受理できなくなる．この L を $s_2(n)$ 領域で受理する万能 TM を設計することで，領域階層定理が証明される．時間階層の証明も同様である．

5-2-2 非決定性領域量の階層性と詰め込み論法

対角線論法で，非決定性領域量の階層定理を証明しようとすると，Savitch の定理で非決定性 TM を決定性 TM で模倣する必要がある．そのため，$\inf_{n\to\infty} \frac{s_1(n)^2}{s_2(n)} = 0$ を満たす $s_1(n), s_2(n)$ に対して，NSPACE($s_1(n)$) $\subsetneq$ NSPACE($s_2(n)$) という稠密でない階層が導けるにすぎない．

---

* $\inf_{n\to\infty} f(n)$ は，$f(n), f(n+1), f(n+2), \ldots$ の値の最大下界の $n \to \infty$ としたときの極限である．関数 $s(n)$ が領域構成可能（時間構成可能）であるとは，長さ n のある入力列に対して，テープ上のちょうど $s(n)$ 個のセルを使う TM が（ちょうど $s(n)$ 回の動作を行う TM が），すべての $n \geq 1$ に対して存在するときをいう．

詰め込み論法では，NSPACE($s_2(n)$) ⊆ NSPACE($s_1(n)$) という，低いクラス間の包含関係の成立を仮定すると，関数 $f(n) \geq n$ に対して，NSPACE($s_2(f(n))$) ⊆ NSPACE($s_1(f(n))$) という高いクラス間の包含関係が成立してしまうという移行補題を利用する．移行補題では，任意の言語 $L \in$ NSPACE($s_2(f(n))$) に対して，$L[f] = \{x2^{f(|x|)-|x|} \mid x \in L\}$ という言語を定義し，$L[f]$ と L が，NSPACE($s_2(n)$) と NSPACE($s_1(f(n))$) にそれぞれ属することを証明する．ここで，$2^{f(|x|)-|x|}$ は，列 $x \in \{0,1\}^*$ を延ばして入力長を $f(|x|)$ にする「詰め物」である．

移行補題を利用した証明では，関数 $f(n) \geq n$ を，うまく設定して，高いクラス間の包含関係を連続的に導出することが証明の鍵になる．例えば，任意の正整数 s, t に対して，NSPACE($n^{s/t}$) ⊊ NSPACE($n^{(s+1)/t}$) を証明するには，NSPACE($n^{(s+1)/t}$) ⊆ NSPACE($n^{s/t}$) を仮定し，$f(n) = n^{(s+i)t}$, $i = 0, 1, 2, \ldots$ と設定して，複数の高いクラス間の包含関係を導出・連結し，上記の $\inf_{n\to\infty} \frac{s_1(n)^2}{s_2(n)} = 0$ に対する階層性を崩壊させることで矛盾を導く[4]．詰め込み論法は，PRAM や交代性 TM, 一様論理回路族に基づく階層定理の改善にも使われている[5]．

5-2-3 非決定性時間量の階層性と再帰的詰め込み論法

非決定性の時間量に関しては, Savitch の定理に相当するものが知られておらず, 領域量ほど単純には階層性が導出できない．Seiferas らは，再帰的詰め込み論法[7]により，$\inf_{n\to\infty} \frac{t_1(n+1)}{t_2(n)} = 0$ を満たす関数 $t_1(n), t_2(n)$ に対して，NTIME($t_1(n)$) ⊊ NTIME($t_2(n)$) を証明している．

本論法では，任意の帰納言語を L とし，L を受理する決定性 TM M の動作時間を $t(n)$ とする．次に，$e_0x \in L(M) \Leftrightarrow x \in L(M_{e_0})$ を満たす非決定性 TM M_{e_0} を考える．ただし，$L(M)$ は M が受理する言語を表す．入力列 e_0x2^k に対して，再帰的に詰め込みを行う TM M' は，(1) $t(|x|) < |x2^k|$ なら，M が x を受理するという条件の下で M' を受理状態で停止させ，(2) $t(|x|) \geq |x2^k|$ なら，非決定的に選んだ $k' > k$ に対して入力列 e_0x2^k を $e_0x2^{k'}$ に延ばして，万能 TM U を動作させる．すると，M' は，$t(|x|) < |x2^k|$ を満たすまで，再帰的に入力列を延ばしていく．NTIME($t_2(n)$) ⊆ NTIME($t_1(n)$) を仮定すると，再帰的な加速により，受理時間に制限のない任意の帰納言語 L が，$t_1(n)$ 時間で受理されるという矛盾を生じる[7]．

再帰的詰め込み論法を最初に提案したのは Cook[1]であり，任意の有理数 $r \geq 1$, 定数 $\epsilon > 0$ に対して NTIME(n^r) ⊊ NTIME($n^{r+\epsilon}$) が示されている．Seiferas らの論文[7]は，その手法を一般化したものである．また，Seiferas の連作論文[6]は，階層定理のサーヴェイである．

■参考文献

1) S.A. Cook, “A hierarchy for nondeterministic time complexity,” J. Comput. Syst. Sci., vol.7, pp.343-353, 1973.
2) J. Hartmanis, P.M. Lewis, II, and R.E. Stearns, “Hierarchies of memory limited computations, IEEE Symp. on Switching Circuit Theory and Logical Design,” pp.179-190, 1965.
3) J. Hartmanis and R.E. Stearns, “On the computational complexity of algorithms,” Trans. Amer. Math. Soc., vol.117, pp.285-306, 1965.
4) O.H. Ibarra, “A note concerning nondeterministic tape complexity,” J. Assoc. Comput. Mach., vol.19, pp.608-612, 1972.
5) C. Iwamoto, N. Hatayama, Y. Nakashiba, K. Morita, and K. Imai, “Translational lemmas for DLOGTIME-uniform circuits, alternating TMs, and PRAMs,” Acta Inform., vol.44, no.5, pp.345-359, 2007.

6) J.I. Seiferas, “Techniques for separating space complexity classes,” J. Comput. Syst. Sci., vol.14, pp.73-99; “Relating refined space complexity classes,” J. Comput. Syst. Sci., vol.14, pp.100-129, 1977.
7) J.I. Seiferas, M.J. Fischer, and A.R. Meyer, “Separating nondeterministic time complexity classes,” J. Assoc. Comput. Mach., vol.25, no.1, pp.146-167, 1978.

■6 群 - 2 編 - 5 章

5-3 多項式階層

（執筆者：相田　慎）[2009 年 2 月 受領]

5-3-1 階層定理と多項式階層

はじめに，前節の階層定理で述べられた階層と，本節で述べられる多項式階層の違いについて，直観的に説明する．

次の二つの時間計算量のクラス DTIME(n^2) と DTIME($n^{2.00000000001}$) を考えよう．これらを規定する二つの多項式 n^2 と $n^{2.00000000001}$ の差異は漸近的にも明らかにわずかであるが，階層定理により，前者は後者に真に包含される．階層定理は，「ミクロの観点」から各計算量クラスを眺めると，様々なクラスが地層のようなきめ細やかな包含関係を成していて，かつ確かな厚み（真の包含関係）があることを主張している．

一方，既存の計算量クラスを用いて新しい計算量の階層を「マクロの観点」から構築する手法がある．本節では，そのような階層で最も有名な **多項式階層** *(Polynolial Hierarchy)* PH[3] について紹介する．詳細については，荻原[1]や Papadimitriou[2]なども参照して欲しい．

5-3-2 オラクルチューリング機械とオラクルつき計算量クラス

多項式階層の定義に必要な概念として，オラクルチューリング機械を定義する．与えられた**オラクル**と呼ばれる言語 $A \subseteq \{0,1\}^*$ に対する**オラクルチューリング機械** M^A とは，通常のチューリング機械〔本編 4 章 4-1 参照〕の作業テープの他に，オラクルテープと呼ばれるもう一つ別のテープが付随しており，特別な状態 q_{query} を持つ．制御部内の状態が q_{query} に遷移したとき，オラクルテープ上の文字列がオラクル A に含まれるか A へ問い合わせられる．

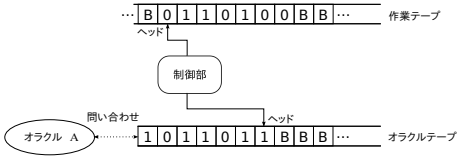

図 5･1　オラクルチューリング機械 M^A

例えば，図 5･1 のような状況で，制御部内の状態が q_{query} に遷移したとき，オラクル A に対して，オラクルテープ上の文字列（図 5･1 の場合は “1011011”）が A に含まれるかどうかを問い合わせをし，1 ステップで結果を得て，それを用いて計算を再開する．

C や Java などのプログラミング言語に詳しい読者にとっては，「オラクル A へのある文字列の所属に関する問い合わせ」は，一見すると「所属するかどうかを高速に計算する関数へのサブルーチンコール程度のもの」に思える．しかしながら，オラクル A は言語であれば何でもよい．つまり，オラクル無し機械では計算できない言語や，決定不可能な言語でもかまわない．そのため，「サブルーチン」ではなく「オラクル」と言う，少々大仰な表現を用いるのである．一般に，オラクル自体の計算可能性や計算量を考慮せずに，オラクルつき計算の計算可能性や計算量を分析する手法を**相対化** *(Relativization)*〔本章 5-1 参照〕と呼ぶ．

次に，与えられた言語クラス $\mathcal{C}$ に対して，いくつかのオラクルつき言語クラスを定義する．

$$\begin{aligned}
\mathrm{P}^{C} &= \bigcup_{A\in C}\left\{ L(M^{A}) \mid M^{A} \text{ は決定性多項式時間オラクルチューリング機械}\right\},\\
\mathrm{NP}^{C} &= \bigcup_{A\in C}\left\{ L(M^{A}) \mid M^{A} \text{ は非決定性多項式時間オラクルチューリング機械}\right\},\\
\text{co-NP}^{C} &= \bigcup_{A\in C}\left\{ L \mid \overline{L} \in \mathrm{NP}^{C}\right\}.
\end{aligned}$$

上述クラスの言語計算では，（非）決定性計算過程でいつでもオラクルへ問い合わせられる．

5-3-3 多項式階層

多項式階層で用いられる言語クラス Σ_n^p, Π_n^p, Δ_n^p $(n \geq 0)$ * を以下のように再帰的に定義する．

0 レベル階層 ：$\Sigma_0^p = \Pi_0^p = \Delta_0^p = \mathrm{P}$.

k **レベル階層** ：整数 $k \geq 1$ に対して，$\Sigma_k^p = \mathrm{NP}^{\Sigma_{k-1}^p}$, $\Pi_k^p = \text{co-NP}^{\Sigma_{k-1}^p}$, $\Delta_k^p = \mathrm{P}^{\Sigma_{k-1}^p}$.

これらを用いて，**多項式階層** PH を次のように定義する．$\mathrm{PH} = \bigcup_k \Sigma_k^p$ $(= \bigcup_k \Pi_k^p = \bigcup_k \Sigma_k^p)$.

Σ_n^p, Π_n^p, Δ_n^p $(n \geq 0)$ ，ならびに PH の包含関係は，「オラクルが付与されたクラスは，少なくともオラクル無しクラスより小さくなることはない」という事実などより，以下のような階層的包含関係になることが知られている[3)]．

定理 1 (多項式階層の包含関係)

1. $\Delta_1^p = \mathrm{P}$.
2. $\Sigma_0^p \subseteq \Sigma_1^p \subseteq \Sigma_2^p \subseteq \cdots$, $\Pi_0^p \subseteq \Pi_1^p \subseteq \Pi_2^p \subseteq \cdots$, $\Delta_0^p \subseteq \Delta_1^p \subseteq \Delta_2^p \subseteq \cdots$.
3. $k \geq 1$ に対して，$\Delta_k^p \subseteq (\Sigma_k^p \cap \Pi_k^p) \subseteq (\Sigma_k^p \cup \Pi_k^p) \subseteq \Delta_{k+1}^p$.
4. $\mathrm{PH} \subseteq \mathrm{PSPACE}$.
5. $k \geq 1$ に対して，(a) $\Sigma_k^p = \Pi_k^p$ ならば $\mathrm{PH} = \Sigma_k^p = \Pi_k^p$, (b) $\Sigma_k^p = \Delta_k^p$ ならば $\mathrm{PH} = \Sigma_k^p = \Pi_k^p = \Delta_k^p$.
6. NP = co-NP ならば，PH = NP = co-NP.

上記の包含関係が真の包含関係になるかについては未解決問題である．定理 1.4 ぐらいは解決できそうであるが，実際は（$\mathrm{P} \neq \mathrm{NP}$ よりも弱い）$\mathrm{P} \neq \mathrm{PSPACE}$ でさえも未解決であることから，その証明は難しいと考えられている．

また，定理 1.5, 6 は多項式階層の**潰れ** *(collapse)* を意味する定理である．例えば，ある自

* ここで，記号として “Σ” や “Π” を用いている理由は，非決定性計算の定義からの類推による．つまり，NP の場合は「ある非決定性計算パスが受理状態に到達可能」つまり**論理和**によるものである一方で，co-NP の場合は「どんな非決定性計算パスも受理状態に到達可能」つまり**論理積**によるものであるからである．また，上付き添字 “p” は polynomial（多項式）の “p” である．

然数 k に対して $\Sigma_k^p = \Delta_k^p$ ならば，PH が Σ_k^p や Π_k^p の定数 k 段の階層で抑えられる（潰れる）ことを意味する．しかしながら，実際にはそのようなことはない（つまり $\Sigma_k^p \neq \Pi_k^p \neq \Delta_k^p$）であろうと考えられている．定理 1.6 は，潰れの極端な例である．

5-3-4 多項式階層の述語論理による特徴づけ

多項式階層を，P に属す言語と述語論理の存在記号 (∃) や全称記号 (∀) を用いることで，別の側面から**特徴づけ** *(characterization)* できる[3]．

定理 2 (多項式階層の述語論理による特徴づけ) 自然数 $k \geq 0$ について，以下が成り立つ．

1. $L \in \Sigma_k^p$ に対して，多項式 $p_1(n), \ldots, p_k(n)$ と言語 $A \in \mathrm{P}$ が存在し，すべての $x \in \{0,1\}^*$ に対して以下が成り立つ:

$$x \in L \iff \exists y_1 \in \{0,1\}^{p_1(|x|)} \forall y_2 \in \{0,1\}^{p_2(|x|)} \cdots Q_k y_k \in \{0,1\}^{p_k(|x|)} \left[(y_1, \ldots, y_k) \in A\right].$$

   ここで，k が奇数のとき $Q_k = \exists$ であり，偶数のときは $Q_k = \forall$ である．

2. $L \in \Pi_k^p$ に対して，多項式 $p_1(n), \ldots, p_k(n)$ と言語 $A \in \mathrm{P}$ が存在し，すべての $x \in \{0,1\}^*$ に対して以下が成り立つ:

$$x \in L \iff \forall y_1 \in \{0,1\}^{p_1(|x|)} \exists y_2 \in \{0,1\}^{p_2(|x|)} \cdots Q_k y_k \in \{0,1\}^{p_k(|x|)} \left[(y_1, \ldots, y_k) \in A\right].$$

   ここで，k が奇数のとき $Q_k = \forall$ であり，偶数のとき $Q_k = \exists$ である．

この定理は，NP や co-NP の特徴づけの一般化に他ならない．式中に ∃ や ∀ が交互に出現することが階層性を暗示しているのみならず，オラクルを用いない階層定義可能性を意味する．

5-3-5 ノート

多項式階層は，計算可能性理論における Kleene の**算術的階層** *(Arithmetical Hierarchy)* のアナロジーとして，Stockmeyer [3] によって定義された．算術的階層とは，決定可能言語クラス REC と帰納的可算言語クラス RE を，オラクルチューリング機械を用いて多項式階層と同様の方法で定義される階層である．特に，算術的階層における REC と RE は，多項式階層における P と NP へそれぞれ対応づけられ，定理 2 のような述語論理による特徴づけ定理も成り立つ（具体的には，定理 2 における P を REC にし，∃ や ∀ で限定された各変数 $y_1, \ldots, y_k$ の長さを制限しなければよい）．更に，本編 4 章 4-5 の定理 2 より，帰納的可算であるが決定不能な言語が存在するため RE ≠ REC となり，算術的階層は階層間に真の包含関係があることがわかる．一方，多項式階層の場合は，本章 5-1 で述べられているように，P ≠ NP が証明されていないため，各階層全てが真の包含関係を持つかどうかは重要な未解決問題である．

多項式階層は，その定義や定理を見てもわかるように，一見すると「複雑極まりないクラス」とも思える．しかし実際は，本編 6 章 6-3 の乱択計算理論や本編 7 章 7-8 の計数計算理論など，様々な計算量の解析に広く用いられる有用な道具であり，非常に重要な概念である．

■参考文献

1) 荻原光徳, “複雑さの階層,” 共立出版, 2006.
2) C. Papadimitriou, “Computational Complexity,” Addison-Wesley, 1994.
3) L.J. Stockmeyer, “The polynomial-time hierarchy,” Theor. Comput. Sci., vol.3, pp.1-22, 1976.

■6 群 - 2 編 - 5 章

5-4 領域計算量

（執筆者：天野一幸）［2009 年 1 月 受領］

様々な計算問題の複雑さを考えるとき，計算にかかる時間と並んで重要なのが，その問題を解くのに必要なメモリ量である．問題のスケールや取り扱うデータ量が加速度的に増加する現在，使用するメモリ量を考慮に入れたアルゴリズムの開発はますます重要になりつつある．これら二つの尺度は，それぞれ時間計算量，領域計算量として定式化され，計算量の解析において最も重要とされる．本節では，領域計算量のうち，特に重要な多項式領域，対数領域といった概念と，領域計算量に基づいて定められる様々な計算量クラスの相互関係について概説する．なお，本稿では専ら，答えとして Yes/No だけを問う決定問題について考える．もちろん，以下に述べる概念は，ある条件を満たす数値や文字列を答えるといった，より一般的問題へと自然に拡張できる．より詳しくは，文献 1, 2, 3) などの優れた教科書を参照されたい．

5-4-1 多項式領域：クラス PSPACE

入力の長さ n に対して，n の多項式で抑えられる量のメモリを使用して解くことのできる問題のクラスを多項式領域（Polynomial-Space）と呼び，通常 PSPACE と表す．厳密には，チューリング機械〔本編 4 章 4-1 参照〕でその問題を解くのに必要なテープ長に基づいて定義される場合が多いが，通常用いられる計算機で使用されるメモリ量に基づいた定義と等価である．クラス P の非決定性版としてクラス NP が定義されるのと同じ要領で，非決定性多項式領域（Nondeterministic Polynomial-Space），クラス NPSPACE も定義することができるが，$f(n)$ 領域を用いた非決定性の計算は $f^2(n)$ 領域を用いた決定性の計算に変換できることを謳う Savitch の定理により PSPACE に等しいことが示されるため，通常用いられない．

PSPACE は相当難しい問題までをも含んでいると考えられている．例えば，NP ⊆ PSPACE である．これは，多項式時間内に停止する機械では，多項式程度のメモリ量しか原理的に使用できないことから導かれる関係 NP ⊆ NPSPACE により明らかである．また，対話型証明系〔本編 4 章 4-5 参照〕で証明可能な問題のクラスは，PSPACE に等しい[4)]．

クラス PSPACE についてもクラス NP と同様に，そのクラスの難しさを代表する完全問題が定義される．それ自身クラス PSPACE に属し，かつ，PSPACE に属する任意の問題がその問題に多項式時間帰着可能な問題を，PSPACE 完全と呼ぶ．PSPACE 完全問題も多数知られるが，代表的なものは以下に示す QBF（量限定ブール式: Quantified Boolean Formula）問題である．論理変数，及び，定数 0 または 1 を論理演算 ∧，∨，¬ で結んでできる式を論理式という．全称記号 ∀，及び，存在記号 ∃ により，すべての変数が束縛されている状態にある式を QBF と呼ぶ．例えば，

$$\exists x_1 \forall x_2 \exists x_3 \{(x_1 \vee x_2) \wedge (x_2 \vee x_3) \wedge (\overline{x_2} \vee \overline{x_3})\}$$

は QBF であるが，この式からいずれかの全称または存在記号を取り除いた式は QBF ではない．QBF 問題とは，QBF が与えられたときにそれが真か偽かを決定する問題である．また，例えば二つの正規表現の等価性判定や，$n \times n$ の大きさの盤面上に与えられたオセロゲームの任意の局面から，白勝利か黒勝利かを判定する問題[5)]なども PSPACE 完全である．

5-4-2 対数領域：クラス L と NL

入力の長さ n に対して，$\log n$ に比例したメモリ量の使用のみ許した機械で解くことのできる問題のクラスを対数領域（Logarithmic space）と呼び，通常 L と表す．ここでは，入力をすべて記憶するのに十分なメモリ量がないので，入力は特別な読込み専用のテープにより与えられるものとし，計算に際して要するメモリ量のみを考える．DVD-ROM 上に書かれた超大なデータに対する解析を，限られた主記憶容量をもつコンピュータを用いて行うことをイメージすると分かりやすい．

対数領域限定の機械は，$2^{O(\log n)}$ 通りの内部状態しかもてないことから，クラス L は多項式時間計算可能な問題のクラス P に含まれることが直ちに導かれる．一方，例えば，n までの値を保持できるカウンタを一つ使用するだけで既に $\log n$ ビットを要するから，対数領域の制約は非常に強いようにも思えるが，クラス L は様々な興味深い問題を含んでいる．例えば，二つの文字列の等価性判定や，正しい括弧列の判定（例えば “(()())()”は正しいが，“)(”は正しくない）など，また，整数上の四則演算も対数領域で計算可能である．最近，Reingold[6] が無向グラフの *st* 連結性問題，すなわち，無向グラフ G と，2 頂点 s と t が与えられたときに s と t が連結であるかを問う問題がクラス L に属することを，拡張グラフと呼ばれる概念を巧妙に用いて証明し，20 年来の未解決問題を解決する画期的成果として話題を呼んだ．

クラス L の非決定性版は通常 NL（Nondeterministic Logarithmic-Space）と表される．定義より明らかに L ⊆ NL であり，また，前節で述べた Savitch の定理より，クラス NL は L^2，すなわち，$O(\log^2 n)$ 領域限定のクラスに含まれる．クラス NL における完全問題として代表的なのが，有向グラフの *st* 連結性問題である．この問題が無向グラフの場合と同様にクラス L に属するか否かは，L=NL かを問うことに等しく，これは領域計算量における重要な未解決問題の一つである．

5-4-3 クラス間の包含関係

上であげたクラスに，いくつかの時間計算量クラスを含めてその包含関係をまとめると以下のようになる．

$$L \subseteq NL \subseteq P \subseteq NP \subseteq PSPACE \subseteq EXPTIME$$

ここで，EXPTIME は指数時間で計算可能な問題のクラスを表す．Savtich の定理と，時間及び領域に関する階層定理〔本編 5 章 5-2 参照〕より，NL ⊊ PSPACE，及び，P ⊊ EXPTIME である．つまり，先頭四つのうち少なくとも一つ，及び，後ろ三つのうち少なくとも一つの包含関係が，真部分集合の関係にある．実際には，上に示したクラスはすべて異なると予想する向きが多いが，現在のところ，これ以外の関係についてはすべて未解決である．

■参考文献

1) S. Arora and B. Barak, “Complexity Theory: A Modern Approach,” Cambridge Univ. Press, 2009.
2) O. Goldreich, “Computational Complexity: A Conceptual Perspective,” Cambridge Univ. Press., 2008.
3) C.H. Papadimitriou, “Computational Complexity,” Addison-Wesley Pub., 1994.
4) A. Shamir, “IP=PSPACE,” J. ACM, vol.39, no.4, pp.869-877, 1992.
5) S. Iwata and T. Kasai, “The Othello Game on an $n \times n$ Board is PSPACE-Complete,” Theoret. Comp. Sci., vol.123, no.2, pp.329-340, 1994.
6) O. Reingold, “Undirected Connectivity in Log-space,” J. ACM, vol.55, no.4, Article 17, 2008.