基礎・一般
解説

計算下界の解明—その意義とシナリオ

〈前編〉

徳山　豪
東北大学情報科学研究科

計算下界の解明の目的

❒人間の思考と計算機の役割

「我思う，ゆえに我あり(cogito ergo sum)」は，思考による真理の追究と人間の存在価値にかかわるデカルトの有名な言葉である．一方「汝計算す，ゆえに汝あり」であるコンピュータは，その高速な計算能力により人間の思考を補佐する.「思う我」と「計算する汝」がどうかかわるかは，情報社会での基本的な問題である.

真理の追究の難しさを計算量の観点から解析し，「思う」に関係した考える，学ぶ，教える，対話する，閃くといった人間の知的活動の価値を，計算の世界で捉えて，どこまでコンピュータがそれを担えるのかを測る．それが計算下界，つまり計算の限界の解明の研究目的である.

コンピュータの力に限界がなく，どんな難題でも解決してくれる未来社会というのは，SF 小説の世界では定番の設定である．そんな夢のような未来は来るのだろうか？ もしかするとそれは夢の未来ではなく，人間の思考の出番も尊厳もない，夢が忘れられた未来かもしれない．計算下界の解明から得られる知見を基盤にして，我々人間の「思う」とコンピュータの「計算」の必須の協力によって難題を解決していく．そこに情報処理の「夢の未来」の姿がある.

❒情報社会の前進への意義

コンピュータが問題を解いてくれたとして，それが正しいかどうか分からないと，多くの場合システムの設計者は説明に困ることになる．したがって，情報技術者はいろいろな現実問題を「解答を見ればそれが正しいことを容易に検証できる」問題として定式化する．次章以降に説明するが，このような問題を専門的には NP[☆1] 問題 (非決定性多項式時間問題) と呼ぶ．ところが常識的には，自力で問題の解答を見出すことは，解答の検証より難しい．科学や文明全般に，偉大な発見にはインスピレーションと想像力，そして運も必要である．つまり「発見は検証より難しい」と考えられている.

この哲学的な問いを定式化して，数理的な問題解決において「解の発見は検証より本質的に難しい」ということを言い切るのが，計算下界の研究課題の代表選手である P≠NP 予想[☆2] である．この予想の傍証として，NP 問題をスパッと完璧に解く方法は現在まで知られていない.

一方で，スパッとは解けないにもかかわらず，コンピュータの計算力を最大限に利用しながら，完璧な答えの出ない諸問題に対して，我々情報科学者はあくなき追及をしていく. 本年度から新学術領域「多面的アプローチの統合による計算限界の解明」が発

☆1 Nondeterministic Polynoimal の略．通常は，非決定性チューリング機械で多項式時間で答えが出る問題として定義されるが，本稿では検証による定義を使う.

☆2 定義を最初に与えるとかえって分からなくなるので，感覚的な意味から入り，後に正確に定義する.

足し，問題解決の追及への指針や手法を見出すためにも，計算の難しさの本質と限界を解明する，我が国の研究潮流の大きな動きが生まれている．

本稿では，社会や科学技術への影響の説明，あるいは哲学的な解釈を加えながら，計算下界に取り組む意義を中心に解説し，研究者の挑戦の現状を紹介する．前編では，計算量や計算クラスを考える動機や意味について解説し，後編では近代的なシナリオや未来への展望，世界と日本の研究動向などについてお話ししようと思う．なお，前編の内容については，標準的な教科書[1]および，上記新学術領域（愛称 ELC プロジェクト）の Web サイト[2]にある渡辺治領域代表の講義録（本稿に内容および画材を提供いただいている）に，より専門的な記述がある．

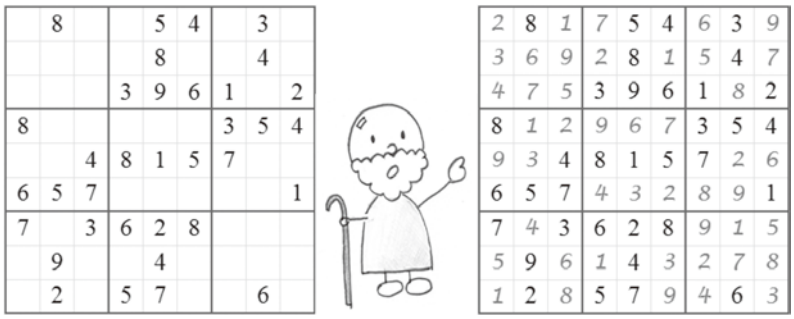


図 -1　数独の問題図と解答図（http://www.websudoku.com/ より）

検証と求解

❒パズルはなぜ面白い

筆者は囲碁，チェス，コントラクトブリッジなどのゲームが好きなので，海外に行くと本屋のゲーム本のコーナーに立ち寄ることが多い．ところが最近では，特に空港などではこのコーナーを独占している本がある．数独（SUDOKU，**図 -1**）である．日本が生み出した大ヒット商品の１つであることに間違いはない．

なぜ数独は面白いのだろうか．まず，第１に，理解しやすく，だれにでもとりかかれるパズルである．第２に，数独には誤魔化しがない．解答が求まれば，それが正しいかどうかはすぐに確かめられる（これを解の検証と呼ぶ）．第３に，ある程度数字が埋まっていくと，選択肢が減ってきて，面白いように解が求まっていく．つまり，空いているマスの数が少ないと，それほど難しくない問題になる．第４に，いくつかの手筋があり，それらを利用するとだんだん空白を少なくしていくことができる．第５に，難易度の高い問題では，それらの手筋だけでは答えが求まらず，メモが必要で，経験者でないと解けなくなってくる．つまり，難問では人間の学習効果がてきめんに発揮される．第６に，でも，どんな数独でも必ずすらすら解けるというマニュアルはいまだに出版されていない（もしもそんなのがあったら，あとは機械的に解けるので，面白くない）．ここで注目する性質は，

「解答を示されれば正しいことはすぐに検証できるが，確実に機械的に解ける方法を示したマニュアルは出版されていない」ということである．

パズルの中でも違ったタイプのものもある．たとえばルービックキューブである．これは初めて手に取るとどうにもならないほど難しく思えるが，「秘訣」（つまりマニュアル）があり，それを知っていればだれでも（スピードに差はあるが）解ける．つまり

「とても難しそうに見えるが，実は隠れた数理的な性質があり，確実に解ける解法を示したマニュアルが作れる」

面白いパズルの条件はいくつかあるだろうが，ほとんどのパズルはこの２つの性質のいずれかを満たしている．我々計算理論の研究者の夢の未解決問題である P≠NP 予想は，この２つの種類のパズルの難しさが本当に異なるということを数学的に示すことである．そしてこれは前章に述べた発見の困難性に一般化される．

情報処理らしく「マニュアル」のことを「アルゴリズム」と言い換えよう．つまり，第１の場合に，「実は良いアルゴリズムが存在するのに，だれも知らないだけ」ではなく，「数理的に効率の良いアルゴリ

ズムの設計が不可能である」ことを示したいのである．

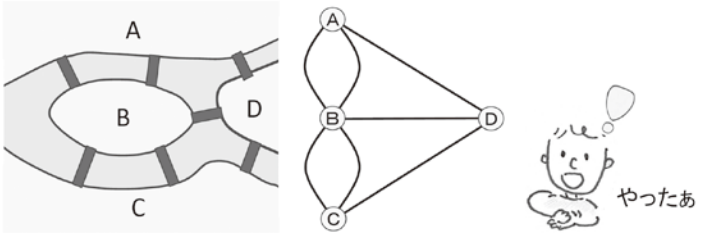


図 -2　ケーニヒスベルグの橋渡り（左図）と対応するグラフ（右図）


図 -3　正 12 面体のグラフ（左図）とその頂点巡礼（右図）

❐ 解ける問題と解けそうな問題

前節のパズルの特徴は数理の本質を表している．数学の問題で答えを見て分からないものは絶望的で，答えを見れば分かるが自分で解くのが難しい問題に醍醐味がある．一方で，大学入試には，確実に解ける方法の示せない問題はなかなか出題できない．そして，問題を見て「これは解ける」「これは難しい」あるいは「これは解答不可能」と自信を持って断言できるのが数学や情報科学，あるいは一般に科学の専門家の真骨頂であろう．

日本の数学の父である高木貞治 (1875 ～ 1960) の数学啓蒙本「数学小景[3)]」を紐解いてみよう．この本は予備知識の要らない問題を題材に，数学の魅力を啓蒙する名著である．出版された昭和 18 年は第二次世界大戦の最中であるが，現代の目から見ると驚くほど計算理論の本質を捉えている．章立てを紹介すると，

(1) ケーニヒスベルグの橋渡り
(2) ハミルトンの世界周遊戯
(3) 隣組，地図の塗り分け
(4) 15 の駒遊び
(5) 魔方陣
(6) 士官 36 人の問題（オイラーの方陣）

これらの問題群はすべて美しく，また解の検証が容易な問題群だが，一般化すると 2 種類に分類される．現代的に言えば，「良いアルゴリズムが存在する（数学的に解ける）」もしくは「良いアルゴリズムの設計が難しい」の 2 種類である．ケーニヒスベルグの橋渡り（**図 -2**）は，すべての橋（図の場合 7 つの橋）を 1 回ずつ渡ってもとに戻る経路を問う問題であり，オイラーの鮮やかな解答で有名である．これはグラフを用いた定式化により，橋がグラフの辺に変換されて一筆書き循環路問題となり，「奇数次数の頂点数が 0 個」であるときにのみ解があることがよく知られている．頂点の次数（接続する辺の数）は簡単に数えられ，ケーニヒスベルグの場合は，次数 5 の頂点があるので，橋渡り循環路は存在しない．つまり，「難しそうに見える（当時の懸賞問題）が，簡潔な数学的判定条件を用いた良いアルゴリズムが存在する」問題であり，解き方の秘訣さえ知っていれば，一目で分かる．

それにひきかえ，ハミルトンの世界周遊戯は，一般化されてハミルトン閉路問題と呼ばれる．これは，「グラフ G が与えられたとき，頂点巡礼（頂点をすべてちょうど 1 回ずつ通る巡回路）を求めよ」という問題であり，特に正 12 面体の頂点と稜線からなるグラフでのハミルトン閉路問題が世界周遊戯（**図 -3**）である[☆3]．高木先生の名文を引用しよう．

> 正 12 面体の頂点巡礼の順路は実は分かっている．それを披露してしまえば，何の変哲もないのだけれども，教わったのでは興味がないから，自分で順路を発見したいものである．それは試行では駄目だから，例の狡猾なる手段を考え出さねばいけない．

「試行」は現代の言葉だと「全探索アルゴリズム」に言い直され，「例の狡猾なる手段」は，「エレガントで効率的なアルゴリズム」と言い直される．つまり

☆3　ハミルトンはこのパズルで一儲けしようとしたが，数独のようには流行らなかった．



図 -4　15 パズル：左図を右図に駒をスライドして直す

ハミルトン閉路問題は答えを見せられれば簡単に納得できる問題である．しかし自分で解こうとすると，全探索アルゴリズムだと大変な手間がかかるので，何か数理的な特徴をとらえて，エレガントで効率的なアルゴリズムを設計しなくてはいけない．

時代に先駆けた明察といえよう．この「例の狡猾なる手段」が一般のグラフの場合でも考え出せるかというのが P≠NP 予想という大問題に繋がるのである．

もう 1 つ取り上げてみると，15 の駒遊びはルービックキューブの 2 次元版のようなパズル（**図 -4**）で，解くための秘訣が知られている問題の例である．一般に良いアルゴリズムができてしまうと，もうパズルとしては面白くなくなってしまう．この 15 の駒遊びは，盤面を大きくしても良いアルゴリズムがある「だれでも解ける問題」であるから，少なくとも大人のパズルとしては面白くないのである．高木先生の名文はこう語る．

一時は熱病のように流行したこの遊戯が，頓に忘れられたのは，それが数学的に解かれてしまったからである．

ということは，もし P≠NP 予想が間違っていて P＝NP ならば，数理的に面白いパズルは激減する（存在しなくなるかもしれない）．パズルマニア諸君には衝撃である．

その他の問題について触れると，オイラーの方陣や魔方陣は，数独の親戚で，いわゆる「制約充足問題」の一例である．地図の塗り分けは有名な四色問題である．これらは「数学小景」で紹介する制限された場合には解けるのだが，一般の制約充足問題やグラフ彩色問題に拡張すると，いわゆる NP 完全問題（定義は後述）になる．

計算階層と計算モデル

❐ 多項式時間計算：取扱い可能な有限

さて，「検証は簡単だが求解が難しい」問題を考えたいのだが，「効率の良いアルゴリズム」というあいまいな表現をより数学的にしよう．

数学では有限と無限の区別は重要である．計算に関しても有限と無限の類別では 1930 年代にチューリングらにより「計算可能性」という概念が確立され，世の中には計算不可能な問題が存在することが知られている．一方で，「有限の時間で計算できる」という判定基準は大雑把すぎて，特に，検証と求解に差を見出すのが難しい☆4．

極端な例を考えてみよう．数学の難問（たとえばリーマン予想）の証明を考えてみよう．証明があるとしたら，それは人間が読んで検証できるものなので，10 メガビット（英語で 125 万字）を超えないであろう（ディジタル写真 1 枚くらいで，データとしてはさほど大きくはない）．夢のような話であるが，もし現代のトップレベルの数学者と同様に「正しく書かれた証明が与えられれば正しいと検証する」ソフトウェアができたとしよう．このとき，超高速なスーパーコンピュータがあったとして，有限時間で自力で証明を与えられ得るであろうか．つまり検証能力を持つソフトウェアは証明能力を持つだろうか？

そんな馬鹿なと思うが，有限時間で計算すればよいのなら，答えは YES である．10 メガビットのテキストは，有限通りの可能性しかない．これらをすべて生成し，上記のソフトウェアでチェックする．証明があれば見つけるし，なければ，「人間に読める証明はない」という回答をコンピュータは与えられる．

もちろん，この手法にはエレガンスのかけらもないが，「検証すること」と「自力で解けること」に差がなくなってしまう．これは実社会感覚とはまったくそぐわない．考えるべきなのは量的な限界であ

☆4　例外的に差が出る場合はある．

る．限りある人生を持つ人間にとっては，あまりに大きすぎる「有限」は「取扱い不可能」になる．10メガビットのテキストは2^{10^7}通りあって，これはどんな未来の高速計算機でも物理的に取扱い不可能な巨大な数である．

つまり，「検証」と「求解」が違うという哲学的な命題を示すには，量的な観点を入れて，高木先生の言葉だと「試行だと駄目」あるいは「狡猾（エレガント）な解法を考え出す」にあたることをきちっと言わないといけない．では，どんなところまでが「効率が良い」計算時間で，どこからが「効率が悪い」計算時間なのだろうか．実用的にも，計算機の計算能力は一定ではなく，一方で取り扱うべきデータサイズもどんどん大きくなる．1970年代だと1ギガ（10億）回の計算をやるプログラムの作動は無理だったが，今はパソコンで動く．したがって，絶対的な値で，「これ以上はダメ」と区切るわけにはいかない．饒舌はこのくらいにして，計算時間の種分けについての標準的な考え方を（教科書レベルで）述べよう．

計算問題には入力サイズというものがある．問題を記述するのに必要なデータサイズが入力サイズであり，たとえばハミルトン閉路問題であれば，おおざっぱに言うとグラフの頂点と辺の数の和がほぼ入力サイズになる．この入力サイズをnとしよう．nが大きいほど通常は問題を解くのが難しくなる．このときに，問題を解くアルゴリズムの計算時間の上界が$f(n)$で漸近的に抑えられるとき，この問題は$O(f(n))$の計算量（complexity）を持つという[☆5]．上界$f(n)$として，定数次数の多項式，たとえばn, n^2, n^{100}などがとれるとき，このアルゴリズムは多項式時間アルゴリズムであると呼ぶ．また，2^nや2^{n^3}などの指数関数（正確には指数部が多項式である関数）で計算時間が抑えられるときには，指数時間アルゴリズムであると呼ぶ．次のように宣言しよう．

定義1　多項式時間アルゴリズムが設計できるとき，この問題は取扱い可能（tractable）であると呼び，そうでないときには取扱い困難（intractable）であると呼ぶ．

n	n^3	2^n（概算）
10	1,000	1,000
20	8,000	1,000,000
30	27,000	1,000,000,000
40	64,000	1,000,000,000,000
70	343,000	1,000,000,000,000,000,000,000
100	1,000,000	1,000,000,000,000,000,000,000,000,000,000

表-1　計算時間と指数爆発

表-1に計算時間の表があるが，$n=10$なら，n^3も2^nもほぼ1000くらいである．ところが，$n=70$くらいでn^3はたいして大きくないが，2^nはスーパーコンピュータ「京」でもなかなか扱えない$10^{21}=100000$京という数になってしまう．先ほど紹介したようにnが10メガくらいになったときに，2^{10^7}などはもってのほかなのである．つまり，指数時間アルゴリズムは現実に実装して動かすと，ちょっと大きいデータでは計算時間が爆発するのである．一方，多項式時間アルゴリズムの計算量はデータ量の増大に対し爆発的には増えない．したがって，問題解決にあたって研究者や開発者がまず念頭に置くべきなのは，指数時間アルゴリズムは極力避け，多項式時間アルゴリズムをどうにかして設計する（設計できないときは問題の定式化から考え直す）ということである．たとえば整列（ソーティング）や，グラフの最短路問題，線形計画問題などには多項式アルゴリズムが設計できるので，それらを用いた定式化が実世界で広く用いられている．

ただし，注意として現実には$O(n^{100})$のアルゴリズムはまったくコンピュータ上で動作させられる代物ではないし，$O(n^3)$でもnが10メガ程度ですら，もう実用的には遅すぎるアルゴリズムである．したがって，上記の「取扱い可能」は実用的には現実と完全には合致しないが，計算時間の大まかな類別としては妥当な区別であると思っていただきたい．

☆5　厳密には，$g(n)=O(f(n))$は，適当な正定数cとn_0をとると$g(n)<cf(n)$が$n\geq n_0$で常に成り立つことを意味し，$g(n)=o(f(n))$は，それ未満，つまりどんな定数cを考えても，十分大きなnでは$g(n)<cf(n)$になることを意味する．また，$g(n)$が$o(f(n))$でないときに$g(n)=\Omega(n)$であるという．

❒休憩：ノーベル賞の話題から

最近テレビで話題の多項式時間アルゴリズム設計のエピソードを紹介しておこう．ノーベル賞を受賞された山中伸弥先生は，有名な山中因子という，細胞の iPS 化を発現する遺伝子集合を発見するときに，数千の因子から 24 の候補因子を選び，それらの組合せをすべて試そうとした．

n 個の候補の組合せは 2^n あり，$n=24$ でも結構大きく，実験には大変な手間がかかる．我々の言葉だと，これは指数時間アルゴリズムであり，「それは試行では駄目だから」ということになる．ところが実験担当の若手研究者は賢い人で，まず 24 個の因子を全部入れてみて発現を確かめ，次に，23 個（つまり 1 個を除いた）の因子の 24 通りの組合せを試してみた．発現しなかった 4 つの組合せで除いた因子が，山中因子の 4 つ組を構成する．

ある程度脚色してあるのだろうが，これは，2^n 回の実験を $n+1$ 回に減らした，我々の言葉だと $O(n)$ 時間のアルゴリズムの設計である．高木先生の言葉だと，「例の狡猾なる手段を考えだした」のである．

❒階層定理

多項式時間アルゴリズムが設計できる問題（つまり取扱い可能な問題）を多項式時間クラス（クラス P）に入る問題であると呼ぶ．さらに，指数時間アルゴリズムが設計できる問題を指数時間クラス（クラス EXP）に入る問題であると呼ぶ．まず最初に生ずる疑問は，この 2 つのクラスが同じであるかどうかである．実際，この 2 つは異なる．

定理 1　P と EXP の 2 つのクラスは異なる．すなわち，指数時間アルゴリズムが設計できるが多項式時間アルゴリズムの設計が不可能な問題がある．

比較対象として無限の世界では，自然数全体の集合の大きさ（可算無限 $\aleph_0$）と実数全体の集合の大きさ（連続無限 $\aleph_1$）には $\aleph_1=2^{\aleph_0}$ という関係があり，$\aleph_0$ と $\aleph_1$ の間の大きさの無限集合はないとしてよい[☆6]．EXP と P でも同じような関係があるのではないかという誘惑に駆られる．

ところが計算階層は，一般の有限集合が「要素数」で区別されるように，下記の階層定理を用いて計算時間（専門的には時間計算量）で定量的に区分される．

定理 2（階層定理）　任意の関数 $f(n)>n$ に対して，$o(f(n))$ では計算できないが $O(f(n)\log f(n))$ では計算できる問題が存在する．

ただし，定理 2 の「問題」は「$o(f(n))$ で計算できる問題すべてと違う答えを出す」天邪鬼のようにひねくれたもので，「対角線論法」を用いて構成する．自然な問題でたとえば $O(n^3)$ で解けて $o(n^2)$ で解けないと証明されているものは見つかっていない．

さて，上の 2 つの定理から，多項式時間クラス P は計算時間に関しては対数的に小さいステップで細かく階層分けされて，EXP はそれより真に大きい．またたとえば $f(n)=n^{\log n}$ とすると，これは多項式関数でもないし指数関数でもないので，P と EXP の間にも中間層があり，それが階段のように階層化されている．ここで問題の難しさの分類と満足してしまえば，それでおしまいである．

しかし，どうもこの階層化は都合が悪い．重要な問題（たとえばハミルトン閉路問題）がどの階層に入っているかというと，とんと分からない．それに，このような定量的な階層化では包丁の入れどころがなく，「計算困難性」という定性的な議論をするためには「使えない階層化」になってしまっている．

そこで，単純な計算時間ではなく，定性的な概念を導入して，計算モデルを用いた階層化を行うことが重要になる．P と EXP の間にはいろいろな計算モデルに関する中間階層があり，重要な計算問題がそこにたくさん詰まっているのである．

❒NP 問題の定義とその位置

現実に世の中で解きたい問題のほとんどは，前節までで述べたように「検証は容易，だけど容易には

☆6　連続体仮説という，数学の公理体系の外にあり，証明は不可能で，正しいとしても正しくないとしても矛盾が生じない不思議な仮説による．

解けない」問題である．定式化すると，「検証が多項式時間でできる」ということになる．少し正確に書こう．ここでは問題は Yes, No を答える問題（決定問題）だけを考えよう．答えが Yes である場合に，Yes という答えに加えて，その答えが正しいことを多項式時間で検証できるような証拠を提出できる，そのような問題を考えよう．

このような問題は非決定性多項式時間クラス（クラス NP）に入ると言い，簡単に NP 問題と呼ぶ．たとえば，ハミルトン閉路問題であれば，「このグラフにハミルトン閉路はありますか？」というのが決定問題で，Yes であるときには，ハミルトン閉路自身を証拠として提出する．数独であれば「この数独は答えがありますか？」という決定問題で，Yes であるときには，数字をすべて埋めた数独の答えを提出する．証拠は答えそのものである必要はない☆7．

ここで大切な注意として，NP 問題の条件としては，答えが No のときには検証ができる必要性はない．したがって，NP 問題の否定は NP 問題であるとは限らない．たとえば「このグラフにハミルトン閉路はないですね？」あるいは「この数独は解けない問題（失題）ですね？」に対して（日本語的な文法で）Yes だったとき，証拠の提示は難しいように思える．このように NP 問題の否定であるものを coNP 問題と呼ぶ．coNP 問題は，答えが No のときには検証が容易にでき，一方答えが Yes のときには検証できなくてもよい．いわば NP と coNP での証拠提出は検事と弁護士のような間柄にあり，NP 問題は「立証する証拠は出せるが，反証する証拠は（必ずしも）出せない」coNP 問題は「反証する証拠は出せるが，立証する証拠は（必ずしも）出せない」．標語的に書くと，P⊆NP∩coNP である．もしかするとこの両辺は等しいかもしれない．これも大きな未解決問題である．

さて，検証が多項式時間でできる問題は，必ず指数時間アルゴリズムで解ける．それは，前に数学の

☆7 専門的になるが，たとえば，「この数は素数ですね？」に対する証拠としては，Pratt の検証というものが利用できる．

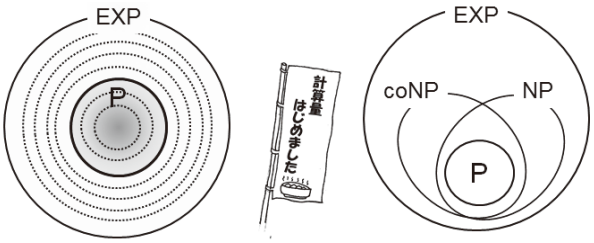


図 -5　階層定理（左図）と，検証モデルによる階層化（右図）

定理証明の話で示したのと同じで，多項式時間で検証できる問題の解答は多項式 $f(n)$ で長さ（ビット長）が抑えられ，そのような解答の可能性は $2^{f(n)}$ しかなく，それを全部試すことが指数時間あればできるのである．

したがって，クラス NP はクラス EXP に入り，一方で NP は P を含む．標語的に書くと P⊆NP⊆EXP である．**図 -5** の右図に構図をまとめてある．

❐ 積み木倒しのシナリオ (1)：NP の内部

ではクラス NP はどのような構造になっているのだろうか？　構造を調べるためには，同じような問題を括って類別するのが通常で，類別には「同値関係」というものを利用する．ここでの同値関係は，「多項式時間還元」というものを利用して定義する．ちょっと数学的になるが，広く知られた重要な概念なので定義を書こう．

定義 2　問題 A が問題 B に多項式時間還元するとは，多項式 f が存在し，問題 A のサイズ n の入力 X が与えられたときに，問題 B のサイズ $f(n)$ 以下の入力 Y を多項式時間で構築でき，問題 A の X に対する答えが Yes であるための必要十分条件が問題 B の Y に対する答えが Yes であるようにできることをいう．

分かりにくい場合は，当面は次の命題だけ覚えればよい．

系 1　問題 A が問題 B に多項式時間還元するとき，問題 B がクラス P に入れば，問題 A もクラス P に入る．

特に，互いに多項式時間還元し合う 2 つの問題は「解ける」ということに関しては同等の難しさ，つ

まり同値だと考えてよい☆8．また，クラスPにある2つの問題は互いに多項式時間還元し，つまり多項式時間還元を通じて，クラスPは1つの問題（同値類）にまとまる．

さて，多項式時間還元を使ってクラスNPを分類するとどうなるのだろうか？　天下りだが，定義を1つしよう．

定義3　AがNP問題で，任意のNP問題XがAに多項式時間還元するとき，AをNP完全問題と呼ぶ．

つまり，もしあるNP完全問題Aが多項式時間で解ければ，すべてのNP問題が多項式時間で解け，P＝NPという（我々の予想に反した）結論が得られる．簡単に言うと，NP完全問題はNP問題のボスであり，このボスを倒すとすべての問題がドミノのように倒れるのである．

一般の集団ではボスは一人あるいは数人である．ところが，NPの世界にはボスはたくさんいる．最初に発見されたNP完全問題は充足問題（SATと呼ぶ）であり，Cookの定理と呼ばれている．その後，S. CookやR. Karpらにより多数のNP完全問題が発見され，ハミルトン閉路問題などを含めた網羅的なリストが作られた[4]．これをNP完全性の理論と呼ぶ．

たとえばハミルトン閉路問題はそれ自体は1つの特殊なマニアックな問題でしかないように思える．したがって，高木先生の頃の昔の数学だと，ハミルトン閉路問題は科学全般には大きな影響がないと思われていた．NP完全性の理論は，このような特定の問題の難しさを完全に解明すれば，PとNPに関する哲学的な問い全般に答えることができることを意味する．

数独もNP完全である☆9から，数独を解く秘伝を開発した場合は，もちろんパズルマニアとしてはすごいことだが，もしその秘伝が多項式時間アルゴリズムであったとしたら，P＝NPを証明したことになる．

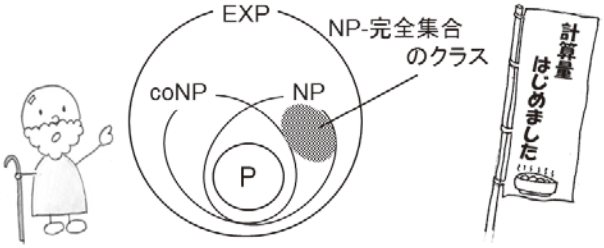


図-6　NP完全問題の位置の可能性

したがって，「数独を解く多項式時間アルゴリズム」はNP完全性の理論を経て，生産工程のスケジューリング問題，タンパク質の構造の最適化問題，物理の多体問題，集積回路の配線最適化，経済均衡の計算，ロジスティクスの最適化など，NP問題として定式化される現代社会の重要問題を一挙に解決する，あるいは大前進させるのである．

閑話休題．NP完全性の理論に対する網羅的な研究から得られた経験則は次のものである．

観察1　クラスNPに入る知られている問題は，可能な少数の例外を除いて，クラスPに入るかNP完全であるかのどちらかである．

図-6の網掛けの部分がNP完全の位置だが，どのくらい大きいかは分からない（もちろんP＝NPなら，NPとcoNP全部がPに統合される）．現状でPに入ることもNP完全であることも証明されていない重要問題としては，自然数の因数分解問題とグラフ同形判定問題（次回登場）が知られている．因数分解において自然数Nの入力サイズはそのビット長$n=\log N$であり，また，「Nが因数分解できるかどうか」という問題，（つまりNが素数かどうかという問題）はクラスPに入り☆10，計算複雑度が未解決なのは実際の因数を求める問題になる．因数分解に関しては，指数時間より本質的に速い計算時間を持つアルゴリズムがあり，また量子計算機を用いれば多項式時間で解ける．これらはNP完全問題とは異なった特徴であるため，因数分解はNP完全ではない可能性が高い．

☆8　ここには落とし穴があり，それについては次回お話しする．

☆9　$n^2\times n^2$に広げて，与えられた表が数独の解の条件を満たすように埋められるかという判定問題がNP完全．

☆10　これは3000年以上人類が考え続けた問題で，2002年にインドの3人の研究者によって解明された[5]．一読を薦める．

NP の上に積み重なる階層

医学でバクテリアの存在を調べるときには，培養によって増殖させ，コロニーを作らせて測定する．また，光学や実験物理学では，微小な光やシグナルを増幅して，計測できるような装置を作り上げる．同様に P≠NP に対する 1 つのアプローチは，P と NP の違いを用いて積み木のように構造物を作り上げ，十分に高く積み上げることによりその違いを明確にしようというものである．

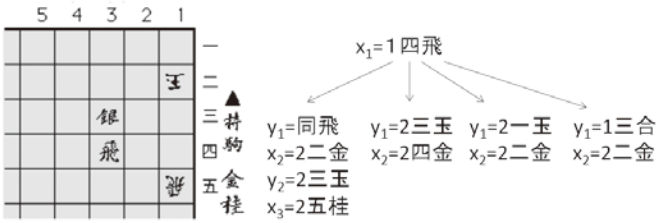


図 -7　5 手詰め詰将棋の論理（詰将棋図式は http://www.geocities.co.jp/playtown/6157/3-5te/gotedume.html より引用）

❒ NP より難しそうな問題

最初に,『情報技術者はいろいろな現実問題を「解答を見ればそれが正しいことを容易に検証できる」問題として定式化する.』と書いた．これは重要なシステム設計指針なのだが，1 つの理想であり，現実世界では解答を検証するのも難しい問題，すなわち NP より難しそうな問題にも取り組む必要が生じる．

代表的なのが，将棋や碁のような対戦ゲームである[☆11]．たとえば，将棋のプロが「この局面は先手が有利である」と言ったとしよう．それをどうやって検証するだろうか．将棋のすべての変化に対して対策を考えると，場合の数は膨大になる（あるプロは 1 秒間に 1 億手以上読むなどと言われる）．したがって，1 つの戦型の主要変化だけでも「四間飛車穴熊必勝法」などという分厚い解説本ができてしまい，それは，将棋の熟達者でないと読んでもちんぷんかんぷんである．その上に，常勝の大名人が書けば説得力があるが，著者の棋士の実戦での勝率が低くなったりすると，全然「必勝法」でないので誰も信じてくれないことになりかねない．

他人事ではなく，実は今書いているこの解説も，すべての人が検証できるわけではなく，本誌を読む読者層を仮定していて，また，筆者への「信用」がきっと大きく読者の理解度を変えるだろう．

閑話休題．「熟達者でないと分からない」「著者の信用や戦法の勝率」についての考察は次回に回し，ここでは，対戦ゲームのモデルを使って，上述の NP の上の積み木構造を作っていこう．

❒ 積み木倒しのシナリオ (2)：NP から組み上げる

対戦ゲームの必勝手順は，「最善手 x があり，それを指す」「相手がどんな手（y）を指してきても」「自分はそれぞれに対して，ある手（最善手）を指す」の繰り返しである．論理記号[☆12]では，「ある手 x」は，$\exists x$「相手のどんな手 y に対しても」は $\forall y$ と書く．

実際の例で詰将棋を考えよう．

$\exists x_1 \forall y_1 \exists x_2 \forall y_2 \exists x_3 Q(x_1, y_1, x_2, y_2, x_3, s, \mathit{Checkmate})$

というのが 5 手詰めの詰将棋の形式である（**図 -7** が例）．詰将棋のルールに基づいて攻撃側の手 x_i は王手で，y_i は王手を解除する手であり，最後の Q の部分は，初期局面 s から開始して最終的な局面が詰み（王手を解除できない状態）であることを示す．

逆に，詰将棋が 5 手で「詰まない」ことを示すには，最初に攻撃側がどんないい手 x_1 を指しても，相手には良い受け手があって詰まないということなので，

$$\forall x_1 \exists y_1 \forall x_2 \exists y_2 \forall x_3 \neg Q(x_1, y_1, x_2, y_2, x_3, s, \mathit{Checkmate})$$

となる．ここで最後の論理式 $\neg Q$ は「詰みではない」ことを意味し，そのほかは $\forall$ と $\exists$ が入れ替わる．

そこで，これにならって，計算問題の難しさを記述するために，論理記号を利用した便利な記法を用意しよう．NP 問題は，Yes の場合は「問題の条件 Q を満たす解 x がある（論理記法では $\exists x, Q(x)$）」，

[☆11] 将棋が分からなくても雰囲気だけ分かっていただきたい．

[☆12] 講義ではその E や A の裏返しはなんですかと聞かれたりする．

No の場合は「任意の x に対して，問題の条件 Q は成立しない（$\forall x, \neg Q(x)$）」の形をしている．これを標語的に（$\exists|\forall$）と書こう（正しいときは存在証拠を示せる．正しくないときはすべてでダメ）．

先ほどの詰将棋だと，1 手詰み詰将棋に対応する．つまり，局面 s が 1 手で詰むのなら $\exists x_1 Q(x_1, s, Checkmate)$ そうでなければ $\forall x_1 \neg Q(x_1, s, Checkmate)$ である．同じ記法を用いると，coNP 問題は，NP の否定なので，（$\forall|\exists$）の形である．

では，NP 問題を 1 手進めてみる．先ほどの記号を拡張して使うと，（$\forall\exists|\exists\forall$）の形の問題である．具体的には，Yes の場合は，$\forall x \exists y Q(x,y)$ という形で，No の場合は，$\exists x \forall y \neg Q(x,y)$ である．

詰将棋の喩えだと，Yes ならば，防御側から開始して，どんな防御の手 x に対しても攻撃側の手 y があって，詰むということである．ただし，x や y は多項式長の選択肢を許すので，その場合の数は指数通りあり，1 手の候補が少ない詰将棋よりも格段に難しい問題まで含める．

このような問題をクラス $\prod_2^p$ に入る問題と呼び，その否定命題であり，裏返した（$\exists\forall|\forall\exists$）の形の問題をクラス $\sum_2^p$ に入る問題と呼ぶ．これらの問題は NP や coNP 問題を 2 段に入れ子に積み上げたものと考えられる．

さらに積み上げて，（$\forall\exists\forall|\exists\forall\exists$）にしたものを $\prod_3^p$，以下同様に $\prod_k^p$, $\sum_k^p$ を定義することができる．これらのクラスの集合和 $\cup_{k\geq 1} \prod_k^p$ と $\cup_{k\geq 1} \sum_k^p$ は等しく，これを（ちょっと紛らわしいのだが）多項式時間階層と呼び，PH と書く．

万一 P＝NP ならば，$\prod_k^p = \sum_k^p = P$，さらに PH＝P となる．つまり一番下の積み木がつぶれると，全体が倒れる．逆に，十分高く階層を積み上げて，本質的にクラス P より難しいものを発見したら，P≠NP になるのである．

❐ PH の行き着く先

多項式計算階層 PH はすべての k に対して $\sum_k^p$ を含むが，この k は定数である．つまり列（$\exists, \forall, ..., \exists, \forall$）の長さ k は定数に限られる．この「定数」というのが曲者である．

初歩的な話で理解を深めよう．ビッグ O 記法について教科書には $O(n)+O(n)=O(n)$ とある．ある再帰的なアルゴリズムの計算時間 $f(n)$ が $f(n) \leq f(n-1)+n$ という関係を持つとしよう．$f(n)=O(n)$ という誤答を帰納法で示そう．$f(n-1) = O(n-1)$ を仮定すると $f(n)=O(n-1)+n=O(n)+O(n)=O(n)$，証明終わり．標準的な学生には誤りが分からない．$O(n)$ を定数回加えると $O(n)$ であるが，n 回加えると $O(n^2)$ になる．ビッグ O 記法で計算量をくくってしまうと，帰納法が成り立たないのである．

では，多項式階層での k が定数でないとどうなるかというと，k が入力の多項式サイズまで増えると，PSPACE[☆13] という新しい 1 つのクラスにまとまる．

先ほどの 5 手詰めの詰将棋は，盤面を 9×9 から $n \times n$ に一般化して考えると，形から見て，これは $\sum_5^p$ に入る．ところが，「$2n+1$ 手詰めの詰将棋」を考えると，これは手数が定数でないので，PSPACE に入る問題である．

PSPACE は PH を含むが，PH より上に上がると，上記の帰納法の破綻と同じように，k を 1 つ増やして階段を上がって，任意の高さまで登っていけるという議論が成り立たなくなる．そのため，たとえ P＝NP であっても P＝PH までは示せるが，階段を PSPACE まで登っていけないので，P＝PSPACE であるとは限らない．

つまり，計算階層が**図 -8** のように組み上げられている．PSPACE は NP の階層の行き着く先の天井（完備化）であるので，筆者を含めて研究者は P≠PSPACE がまず取り組むべき関門で，P≠NP よりも性質はよさそうで，かつ大きな道標になるのではないかと感じている．

注意しておくと，対戦ゲームであっても必ず難しいとは限らない．たとえば，次の石取りゲームを考えて見よう．n 個の碁石がある．二人で交互に 1, 2,

☆13 PSPACE は多項式サイズの記憶領域で解ける問題群として定義され，本稿での定式化は「交代性チューリング機械で多項式時間で解ける」問題が PSPACE と一致することを用いている．

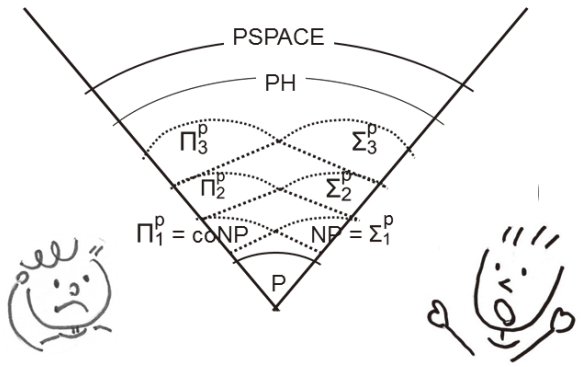


図 -8　多項式時間階層 PH の構造

または 3 個の石を取り除く．最後の石を取らされた方が負けである．石の数が 2013 個なら，先手と後手，どちらが勝つだろうか？　これには簡単な仕掛けがあり，2013 個なら後手必勝である（ご存じない場合は頭の体操によい[6]）．

同じような必勝法が将棋のような複雑なゲームで存在するとは思えない．でも絶対にないと数学的に言い切れるだろうか☆14．これが P≠PSPACE を解明するということの直感的な意味である．

次回予告

今回の内容は 1980 年頃に整備された部分で，その後研究者たちは下界研究を阻む壁に突き当たる．この壁へのアタックのため，オラクルモデル，回路計算量，乱択モデル，対話証明，確率的証明，幾何的複雑度理論など多様なシナリオや道具が開発され，研ぎ澄まされているのが現状である．次回はそれらを手に世界や日本の若い研究者たちが挑んでいく様子をお話ししようと思う．

☆14 囲碁では太閤秀吉が反証を試みた．盤の中央にまず着手し，その後相手の着手の対称点に着手する．太閤真似碁という．

参考文献

1) Sipser, M.（渡辺　治，太田和夫監訳）：計算理論の基礎，共立出版 (1999) [第二版（三分冊）2008]．
2) 多面的アプローチの統合による計算限界の解明 (ELC Web ページ)，http://www.al.ics.saitama-u.ac.jp/elc/
3) 高木貞治：数学小景，岩波書店 (1943)（改定 1981, 2002 岩波現代文庫　学術 81）．
4) Garey, M. R. and Johnson, D. S. : Computers and Intractability — A Guide to the Theory of NP-Completeness (1979).
5) Agrawal, M., Kayal, N. and Saxena, N. : PRIMES is in P, Annals of Mathematics 160-2, pp.781-793 (2004).
6) 一松　信：石取りゲームの数理，森北書房 (2003)（初版 1968）．

(2012 年 12 月 27 日受付)

徳山　豪（正会員）tokuyama@dais.is.tohoku.ac.jp

1979 年東京大学理学部数学科卒業．1985 年同大学院理学研究科数学専攻修了（理学博士）．1986 年日本アイ・ビー・エム東京基礎研究所研究員．1999 年より東北大学大学院情報科学研究科教授．現在，同研究科副研究科長．アルゴリズム理論を中心に理論計算機科学の研究に従事．日本 IBM 科学賞，船井情報科学財団振興賞，本会研究賞，同ベストオーサー賞等受賞．本会，電子情報通信学会フェロー．宮城県囲碁アマチュア本因坊（3 回）．