【解説】

解の個数を数えることの複雑さについて

数え上げ問題の計算量

戸田誠之助
日本大学文理学部応用数学科

はじめに

情報科学の基礎理論の1つに計算量理論（もしくは，計算の複雑さの理論）と呼ばれる分野があります．この分野の最も素朴なテーマは，情報処理上のいろいろな問題を数学的な計算問題として定式化し，その計算問題自身の処理効率（以下，計算量と呼びます）を分析することなのですが，これ以外にも，いろいろな計算構造の性質を調べたり，計算構造間の相互関係を調べたりもしています．

私自身は，計算構造の相関関係を調べることに興味を持って研究を進めてきました．まず，このことが何を意味しているのかを説明したいと思います．そのあとで，私自身が行った研究について手短に述べることにします．

戸田誠之助氏のゲーデル賞受賞について

笠井琢美　電気通信大学情報工学科

ある問題に対し，“解が存在するか?”という問Aと，“解がいくつ存在するか?”という問Bではどちらがどれくらい難しい問題でしょうか? 問Bが解ければ問Aは解けます（解の個数が0のときは解は存在しませんし，解の個数が1以上のときは解は存在します）．逆に問Aが解けても問Bは解けません．したがって，“問Bは問Aより難しい”ということが予想されます．では，問Bはどのくらい難しいのでしょうか? 数年前，この質問に関するある定理を戸田誠之助氏（日本大学文理学部助教授）が発見しました．そして昨年の7月，その発見に対し，理論計算機科学の分野で栄えある賞の1つであるゲーデル賞が授与されました．このことが理論計算機科学のみならず広く情報処理にたずさわる若い研究者への励みになればという思いを込め，ゲーデル賞と戸田誠之助，受賞対象となった研究について簡単に紹介したいと思います．

●ゲーデル賞について　ゲーデル賞は，理論計算機科学の分野で顕著な発見を行い，学術専門誌に発表した論文に対して与えられる賞で，EATCS（ヨーロッパ理論計算機学会）とACM-SIGACT（米国コンピュータ学会のアルゴリズム・計算理論研究会）の主催により，毎年1回受賞者が決定されます．昨年は6回目にあたります．授賞式はICALP（オートマタ・言語・プログラミングの国際会議）とSTOC（計算理論の国際シンポジウム）で1年おきに交互に行われており，昨年はデンマークのアールボークで開催された第25回ICALP（7月13日～17日）の中のEATCS総会で授賞式が行われました．戸田さんはこの授賞式に出席し，記念講演を行いました．この賞の名称は，いうまでもなく，ゲーデルの不完全性定理として有名なクルト・ゲーデルにちなんで付けられたものです．今回の受賞は，戸田さんの論文「PP is as hard as the polynomial-time hierarchy,"SIAM J.Computing 20 (1991)」が評価されたものです．ゲーデル賞を受賞するのは，日本人としては初めてです．彼はこれだけでなく，1989年に開催されたFOCS（計算機科学基礎理論の国際会議）において，発表論文が最優秀論文に選ばれたこともあり，これも日本人として初めてのことでした．彼の業績は，非常に多くの人に影響を与え，多くの論文で引用され，最近では教科書にまで戸田の定理として現れるようになってきました．

●戸田の定理について　多項式時間階層の説明，P = NP問題とのかかわり，を中心とし戸田の定理をごく簡単に説明します．真か偽かを表す表明を述語といいます．n個の変数を含む述語をn変数述語といいます．たとえば“xはyが好きだ”は2変数述語です．これを$P(x, y)$で表すことにします．これに$\exists x$とか$\forall x$を付けると新しい述語ができます．$\exists$と$\forall$は，英語の量限定詞someとallに相当します．たとえば，$\exists yP(x, y)$は“xには好きな人がいる”という1変数述語を表し，$\forall xP(x, y)$は“yはすべての人に好かれている”という述語を表します．こういった量限定詞の付いた述語は，数学や論理学で古くから扱われていますが，特に（古典的な）計算の理論では，次の形の述語を研究してきました．

$$\exists x_1 \forall x_2 \cdots Q x_n P(x_1, \cdots, x_n, y)$$

ここで，Pは計算可能な述語，Qはnが奇数か偶数かに応じて$\exists$か$\forall$を表すものとします．このようにして表される問題のクラスをΣ_nで表します．Σ_0は計算可能な問題のクラスを表します．すると問題の階層

$$\Sigma_0, \Sigma_1, \Sigma_2, \cdots$$

背景

計算量理論の目標は，個々の計算問題の計算量を厳密に分析し尽くしてしまうことです．特に，計算量理論研究者は，個々の計算問題がある計算量以下では処理できない，といった否定的な事実を証明する（すなわち，計算量下界を証明する）ことに強い関心を抱いています．しかしながら，現時点では理論的な基盤が十分ではないために，通常，議論の下地になっている計算モデルを制限することによって計算量下界を証明しています．この目標に関する現在の研究活動は，地道に研究成果を積み上げながら，理論的な基盤を構築していると考えられます．

一方，計算量を分析するために，1970年代から脈々と行われてきているもう1つの方法論があります．それは，計算問題の間の相対的な関係を調べるというものです．つまり，この計算問題はあの計算問題より（計算量という点から考えて）難しいとか，この計算問題とあの計算問題はほぼ同等の計算量を持っているといったことを議論するわけです．言い換えれば，計算問題全体をそれが本質的に備えている計算量に基づいて序列化を行おうというのが，この方法論の趣旨です．NP完全性の理論はこの方法論に基づいた典型的な枠組みといえます．

1970年代から1980年代の前半までは，計算問題そのものを議論の対象としてきました．一方，それらの研究活動を通して，計算問題の相対的な比較分析は，各計算問題の背後にある計算構造を比較していたのだということを認識するようになりました．このため，1980年代の後半には，個々の計算問題そのものを扱うよりも計算構造を議論の対象とするようになっています．

が得られます．これをクリーネの階層といいます．Σ_1は特に重要でこれを帰納的可算集合といいます．

ここで述べた計算可能という概念は，単にアルゴリズムが存在することを意味します．（現在の）計算量の理論（文字“量”が入っていることに注意）では，$P(\cdots)$は単にアルゴリズムが存在するというだけでなく，実際的な時間で計算できる（正確には多項式時間計算可能）という条件を付け加えます．このようにして得られるのが戸田さんが研究対象とした多項式時間階層です．すると，Σ_0はクラスPに対応し，Σ_1はクラスNPに対応します．したがって，P = NP問題は多項式時間階層の根っこにある問題といえます．

戸田さんの業績は（非常に複雑な構造をしていると一般に信じられている）多項式時間階層が，数え上げ問題という（単純で明確な）概念で特徴付けられることを示したことです．数え上げ問題とは，解の個数を問う問題です．たとえば，上で述べた“xには好きな人がいるか”という問題は単に存在を問うだけですが，数え上げ問題では，“xには好きな人が何人いるか”を問うことになります．

●戸田の定理の意義　評価される研究には，新しい製品の発明とか新しいアルゴリズムの開発に結び付かなくても，その研究分野に大きな影響を与えるものが多くあります．例としては，少し大げさですが，NP完全問題の概念とか，ゲーデルの不完全性定理がそうです．戸田さんの研究には，確率アルゴリズムとか，数え上げ問題など計算に関する新しい発見や見地がたくさん含まれています．将来，戸田さんの仕事がP = NP問題の解決になんらかの影響を与えるのかどうかは現時点では分かりませんが，このような積み重ねがなければ，解決に至ることはないと思います．

●戸田氏について　戸田さんの紹介という意味で，特に彼の学生時代のエピソードを書きたいと思います．なぜ私が書くのかというと，私が彼の指導教官であったからです．私が電気通信大学に赴任して最初に受け持ったのが戸田さんで，卒研生はその年は戸田さん1人でした．彼に最初に読ませた本は，グラフアルゴリズムに関する本で，これが実にひどい本で，数行に1つの割合で誤りがありました．誤りも，ミスプリ程度のものだけでなく本質的なものもたくさんありました．それを彼が全部直してくるものですから，ゼミにすごく時間がかかりました．戸田さんは，ゼミの用意が不完全だとすごく怒られたと言っていますが，真相はゼミを早く切り上げてほしかったからだけです．ゼミのとき，私がいつも怒っていたのは，戸田さん以外はほんのわずかな（どういうわけかゼミが好きで，ゼミをするようにしつこく言ってくる）人だけです．不完全な本でも，人によっては非常に良い教育的効果を上げることがあるものです．不完全な本（や不完全な指導教官）だと，自分でもできるという自信を与えてくれることがあるからです．また，いくら勤勉でも，好きな人には勝てません．好きな人は，食事の時も，お風呂に入っている時も，寝ている時さえそのことを考えており，かけている時間が違うからです．

戸田さんがゲーデル賞を受賞したのも，研究が好きで，研究に情熱を持っているからであると思います．彼の受賞は，日本の若い研究者の励みになり，そのことが一番の喜びでもあります．

計算構造の抽出

ここで，「計算構造」という言葉が何を意味しているのかほとんど理解できないことと想像します．計算量理論研究者が抱いている直感的概念（数学的概念ではありません）なので説明するのは困難ではあるのですが，以下では，具体的な計算問題をもとにこのことを大まかに説明したいと思います．

次のような3つの計算問題を考えてみましょう．以下では，行列とベクトルといったときには，0と1（正確には，ガロア体GF(2)の要素）を成分とする行列ならびにベクトルを表していると考えてください．また，ベクトルはすべて横ベクトルのことだと考えてください．以下で使用する$w_H(\vec{x})$という記法は，ベクトル$\vec{x}$の成分1の個数（符号理論でいうところのハミング重み）を表しています．

入力：$m \times n$行列H，n項ベクトル$\vec{s}$，自然数W．

判定：$\vec{e}H^T = \vec{s}$かつ$w_H(\vec{e}) \leq W$を満たすn項ベクトルは存在するか？（注：存在するときにはtrueという答えが，存在しないときにはfalseという答えが出力されるものと考えてください．）

ここで，H^Tは行列Hの転置を表しています．なお以下では，（曖昧にはならないだろうと思うので）行列の大きさやベクトルの成分数に関する記述を省略します．

入力：行列H，自然数W．

判定：任意のベクトル$\vec{s}$に対して，$\vec{e}H^T = \vec{s}$かつ$w_H(\vec{e}) \leq W$を満たすベクトル$\vec{e}$が存在するか？

入力：行列H，ベクトル$\vec{s}$，自然数W．

出力：$\vec{x}H^T = \vec{s}$かつ$w_H(\vec{x}) \leq W$を満たすベクトル$\vec{x}$の個数．

$$(e_1\, e_2\, \cdots\, e_n)\begin{pmatrix} h_{11} & h_{21} & \cdots & h_{m1} \\ h_{12} & h_{22} & \cdots & h_{m2} \\ \vdots & \vdots & & \vdots \\ h_{1n} & h_{2n} & \cdots & h_{mn} \end{pmatrix} = (s_1\, s_2\, \cdots\, s_n)$$

以下，この3つの計算問題を上から順にQ_1，Q_2，$Q_\#$と呼ぶことにします．

さて，これらの計算問題を相対的に比較しようとした場合，これらを直接扱ってもよいのですが，計算量理論では，各計算問題の定義に含まれている論理的な様式のようなものを（述語論理式などを用いて）抽出します．

たとえば，問題Q_1は，行列Hとベクトル$\vec{s}$と自然数Wを引数として，問題の定義のところで述べた判定条件が成立したら真となり，そうでなければ偽となるような（論理学でいうところの）述語として定義し直すことができます．さらに，判定条件そのものも述語として定義し直すことができます．この述語を$R_1(H, \vec{s}, W, \vec{x})$で表すことにしましょう．つまり，$R_1(H, \vec{s}, W, \vec{x})$が真となる必要十分条件は，$R_1$の引数として与えられた行列$H$，ベクトル$\vec{s}$，自然数$W$，ベクトル$\vec{x}$が問題$Q_1$の判定条件を満たすこととなります．このとき，問題$Q_1$（の定義）は次のように表すことができます．

$$Q_1(H, \vec{s}, W) \equiv \exists \vec{x}[R_1(H, \vec{s}, W, \vec{x})]$$

計算構造を抽出するというときの趣旨は，計算問題Q_1と何か別の計算問題を比較しようとする場合，Q_1そのものを直接扱うのではなく，上のような様式を持った計算問題からなる集合（計算問題からなる集合を総称して計算量クラスといいます）を扱おう，というものです．そこで，上のような様式で定義できる計算問題すべてからなる計算量クラスを$\exists \cdot \mathbf{P}$で表すことにします．この記法の中の$\exists$は，「（何かが）存在するかどうか」を判定することが問題Q_1の本質であることを示しており，$\mathbf{P}$はQ_1の定義の基礎となっている述語R_1が「簡単に処理できる」ことを示しています．最後に，問題Q_1と計算量クラス$\exists \cdot \mathbf{P}$を正確に結び付けるため，$\exists \cdot \mathbf{P}$に属するあらゆる計算問題がQ_1に効率よく還元できること（つまりQ_1が$\exists \cdot \mathbf{P}$-完全であること）を示す必要があります．この事実が証明できた段階で計算量理論研究者は，Q_1と$\exists \cdot \mathbf{P}$を（計算量という観点から）同一視してしまいます．つまり，Q_1を分析の対象としても$\exists \cdot \mathbf{P}$を分析の対象としても結局は同じことをしているのだと判断するようになります．実際，Q_1は$\exists \cdot \mathbf{P}$-完全になることが証明されているので，（私などは）両者を完全に同じものと見なしてしまいます．

以上のような考え方をQ_2に対して適用すると（詳しくは述べませんが）$\forall \cdot \exists \cdot \mathbf{P}$と表現される計算量クラスを抽出することができます．またさらに，Q_2が$\forall \cdot \exists \cdot \mathbf{P}$-完全であることもすでに知られているので，両者は同じものと見なすことができます．問題$Q_\#$についても$\# \cdot \mathbf{P}$とでも表現される計算量クラスを抽出することができます．ここで，#記号は「（何かの）個数を数える」といったことを表現していると想像してください．なお，この原稿を書いている時点では，$Q_\#$と$\# \cdot \mathbf{P}$を同一視してもよいかどうかは分かりませんが，おそらく同一視できる

だろうと予想します.

以上が計算構造を抽出するといったことの大まかな意味です.

数え上げ問題の計算量

1980年代の後半頃から先に述べたような計算構造を抽出して，それらの性質を分析したり，相対的な関係を分析するといった研究が行われるようになりました．私はちょうどこの頃に本格的な研究を始めたため，このようなテーマに興味を抱きました.

私は#·**P**で表される計算量クラスとヨ·および∀·を有限回重ねて得られる計算量クラス，たとえば，ヨ·∀···ヨ·**P**などと表される計算量クラス（このような計算量クラスを総称して多項式時間階層と呼びます）を比較してみようと考えました．言い換えると，Q_1やQ_2のような様式を持った計算問題と$Q_{\#}$のような様式を持った計算問題の相対的な関係について検討してみました．その結論をいうと，多項式時間階層に属するあらゆる計算問題は，#·**P**と同一視できる数え上げ問題に効率よく還元できる，というものです．直感的にいうと，（$Q_{\#}$と#·**P**の関係は不明ではありますが，仮に同一視できたとしたら）Q_2のような計算問題は$Q_{\#}$のような計算問題に（効率的な方法によって）書き直すことができる，というわけです.

私の研究の中では，判定条件の計算量を表す計算量クラスとして**P**を使用したのですが，この計算量をどのように設定したとしても，ほとんどの場合について上と同じ結果が成立することが分かっています．つまり，数え上げるという操作は，存在性や全称性を組み合わせた条件を判定する操作に比べて，かなり難しい（逆にいえば，より多くの情報を提供してくれる）ものであるといえます.

エピソード

編集委員会のご要望により，今回の研究成果を得るに至ったエピソードのようなものを述べたいと思います.

上記のような研究を始めたころは，関連する論文が数編しかなく，数え上げの計算構造に対してはあまり深く研究されていませんでした．しかしながら，「数え上げる」という操作は至極自然なものなので，自然に興味を持つようになりました．また，論文数が少ないことは，逆に考えれば，適当なテーマを発掘できる可能性が十分残されているとも考えました.

その一方で，Author-Merlin Gameや対話型証明系に関する結果（proverとverifierとの間で交わされる定数回のinteractionを1回のinteractionに置き換えることができること）に興味を持ち，その結果が成立する本質的な理由を自分なりに再構築してみようとも考えていました．やがて，その本質的な理由が確率型計算構造のswapping property（確率型計算構造と他の計算構造とは，その計算順序を入れ替えることができるという性質）にあることを理解し，さらに，当時Schöingが提案していたBP-operator（確率型計算構造を抽象化した概念）を用いることによって，その理由をより簡潔に記述できるばかりか，確率型計算構造に関するいろいろな性質をある種の代数的な議論を用いて比較的容易に証明できることに気づきました．言い換えると，確率型計算構造の性質を分析するための何か新しい枠組みのようなものを直感的に理解したといえます．特に，この枠組みとValiant-Vaziraniの結果（存在型計算構造は確率型計算構造と数え上げ結果の偶奇性を判定する計算構造とを結合したものに置き換えることができる）を組み合わせることによって，Valiant-Vaziraniより強い結果（多項式時間階層全体に対して彼らと同じ結果が成り立つこと）を示すことができました．以上のようにして，今回の研究の前半部分が得られたわけです.

以上の前半部分だけで論文にしてもよかったのですが（そして，それだけでも論文として認知されただろうとは思いますが），当時は先人の結果をうまく組み合わせただけであるという点を強く意識してしまい，あまり独自性を感じることができませんでした．そこで，前半の結果をもう少し発展させてより独自性のある結果を示そうと考え，前半の研究で得られた結論とある独自の証明手法を用いて後半の結果（つまり，多項式時間階層全体が数え上げ計算の構造に還元できること）を示すことができました.

この研究を始めた当初は，大きな問題を解こうとか未解決問題を解決しようなどとは考えていませんでした．実際，数え上げ計算に関する論文は少なく，何が本質的な課題であるかといったことはまだ議論されていない段階でした．このような状況だったため，上記の結果を発表した当初は，どこに出しても恥ずかしくない結果であることは自覚していましたが，高い評価を受けるとはまったく考えてもいませんでした．今回のような受賞対象になったことを自分自身驚いています.

（平成11年1月4日受付）