

Mathematics and Logic as the Cornerstone

特集

「礎としての数理・論理」

数理・論理と情報学

宇宙の星の謎を探るにしろ、台風の進路予測をするにしろ、ロボットを動かすにしろ、その礎に数学がなければ問題を解決することはできない。しかし、数学とはそもそも人間が知りたいと思う欲求、好奇心から始まった学問であり、芸術に似て、それ自身が人間を感動させる「美」をもつと数学者たちは言う。純粋数学の意義から、世の中に貢献する数学、とりわけ情報学に応用される数学まで、数理・論理の研究の魅力について話を聞いた。



佐藤 健
Ken Satoh
情報学プリンシプル研究系教授

実社会を支える数理論理と数理工学

佐藤 まず、龍田先生の研究分野についてお聞かせください。

龍田 私の研究分野は数理論理です。これは、人間が数学を考えたときの思考方法を、その思考手順を厳密に記述し、対象化することにより分析するという学問。ギリシャ時代の古から人間が探求してきた非常に長い歴史をもつ学問ですが、数十年前からコンピュータへ応用されるようになり、より脚光を浴びるようになりました。とりわけ現在では、ハードウェアの回路、ハードウェアやソフトウェアの検証、ソフトウェアの仕様記述、プログラミング言語の設計、人工知能での推論などの分野において、必要不可欠な学問となっています。

たとえば、ソフトウェアの検証というのは、そのソフトウェアが間違いなく動くかどうかを数学的に証明して確かめるというもの。というのも、ソフトウェアは、いまだに人が手作業で書いて生産するしか方法がなく、必ず間違えるものだからです。一方で、ソフトウェアが間違っていると、パソコンが動かなくなるとか、銀行のオンラインシステムが止まってしまうとか、飛行機や原発などの社会インフラの制御ができなくなるといった具合に、私たちの生活に重大な支障をきたします。これらはすべて、プログラマーが意識しないで間違えて書いてしまったプログラムの部分、「バグ」によって引き起こされるものです。

そこで通常は、テストプログラムを動かして、だい

たい動くようであれば、そのソフトウェアを走らせるという方法がとられています。しかし、テストには漏れがあり、バグを見逃すことで重大な事故が起こることもある。これはソフトウェアの危機とも呼ばれています。そうしたことから、完全にバグのないソフトウェアを生産するために、ソフトウェアを数学的に検証し、どんな場合でも動くという保証が求められているのです。

一方、現状では、この手法はあまり採用されていません。なぜなら、ソフトウェアの数学的な検証には、大変な手間と費用がかかるからです。とはいえ、ソフトウェアは社会インフラを支えるうえで非常に重要な役割を担うものですから、将来的にはすべてのソフトウェアについて数学的に検証されるようになるでしょう。プログラム理論は、新しい種類の数学であり、ソフトウェアに内在する計算の本質を解き明かすことができる深みのある学問であり、興味は尽きることがありません。

佐藤 たとえば、現在、車にはたくさんのコンピュータが搭載されていますが、それが正しく動くかどうかを検証することも、大きな課題になっています。しかも車の場合はリアルタイムに、瞬時に判断し制御する必要がある。そういった場面でも数学論理の役割というのは非常に大きなものがありそうですね。

龍田 そうですね。さらに、数学論理はコンピュータ教育にも役立っている。たとえば、ソフトウェア技術者の考え方の基礎を養う役割を担っています。

TLCA List of Open Problems

<http://tka.di.unito.it/oplca/>

Updated July 11, 2008

Problem # 20 [SOLVED]

Submitted by Mariangela Dezani-Ciancaglini

Date: 2006

Statement. Type theoretic characterisation of hereditary permutations

According to [Bergstra and Klop, 1980, Barendregt, 1984] (Definition 21.2.9) a closed lambda-term M is a *hereditary permutation* if the Böhm tree $BT(M)$ has the following properties:

- the label $\perp$ does not occur in $BT(M)$;
- each variable occurs exactly once;
- the head variable is the first abstracted variable;
- all other variables occur at one level lower than their abstractions.

The problem is to find a type assignment system in which all and only the hereditary permutations get types of a fixed shape. Many other sets of λ -terms have been characterised in this way, see for example [Kurata, 2002, Dezani-Ciancaglini et al., 2005] and the references there.

Solution: Makoto Tatsuta proved [Tatsuta, 2008] that the set of all hereditary permutations is not recursively enumerable, and therefore cannot be characterized by typability in a finitary type-assignment system. On the other hand, there is a system such that a term can be assigned a certain countably infinite family of types in that system if and only if it is a hereditary permutation.

References

[Barendregt, 1984] Barendregt, H. (1984). *The Lambda Calculus. Its Syntax and Semantics*. North-Holland, second edition.

[Bergstra and Klop, 1980] Bergstra, J. A. and Klop, J. W. (1980). Invertible terms in the lambda calculus. *Theoretical Computer Science*, 11:19–37.

[Dezani-Ciancaglini et al., 2005] Dezani-Ciancaglini, M., Honsell, F., and Motokawa, Y. (2005). Compositional characterization of λ -terms using intersection types. *Theoretical Computer Science*, 340(3):459–495.

[Kurata, 2002] Kurata, T. (2002). Intersection and singleton type assignment characterizing finite Böhm-trees. *Information and Computation*, 178(1):1–11.

[Tatsuta, 2008] Tatsuta, M. (2008). Types for hereditary permutations. In *Proceedings of the Twenty-Third Annual IEEE Symposium on Logic in Computer Science, LICS 2008, 24–27 June 2008, Pittsburgh, PA, USA*. <http://research.nii.ac.jp/TechReports/07-010E.html>.

1

図1 解かれた 20 番目の問題

プログラム理論の国際学会は、数学分野の未解決問題として 22 の問題を挙げている。龍田教授はその 20 番目の「全単射に相当するラムダ式を特徴づける型理論を与えよ」という問題に挑戦し、図のように見事に解決した。

ソフトウェアというのは、論理的に書かれ、論理的に実行されるものであり、数学論理を学ぶことで、技術者に必要な基礎的な素養を身につけることができるのです。

佐藤 なるほど。では、次に速水先生、お願いします。

速水 私の研究分野は数理工学です。一言でいえば、実際に解かなければならない問題があった場合、まず数理の言葉に置き換えて、問題を解決したり、解決するための方法論を編み出したりする学問です。なかでも私の専門は、数値計算という分野。自然現象、工学現象、社会現象などを、コンピュータを使ってシミュレートするための基礎技術の開発を行っています。

たとえば、台風などの気象予測や、飛行機的设计のためのシミュレーションには、流体現象を表す方程式を解く必要があります。流体の数学モデル、数理モデルというものです。次にこの方程式を解くわけですが、複雑な現象や複雑な形を扱う場合、手計算では解けないので、コンピュータを使って〈近似的〉に解きます。そこで必要なのが「離散化」というプロセス。離散化とは、連続またはアナログな方程式を、コンピュータが扱えるようなデジタルな式に変換、または近似することです。さらに、離散化された方程式の近似の精度を上げるためには、たとえば、3次元を扱う場合、空間を網目状に膨大なメッシュで切り分ける必要があり、その分、コンピュータで扱う変数の数も膨大になります。したがって、計算機の限られたメモリーと計算速度で答えを

導き出すためには、効率的な計算法(アルゴリズム)を開発する必要があるのです。

また、開発した数値計算法が一般にどう振舞うのか、どんな場合にちゃんと機能するのか、答えの精度はどうか、答えに近づく(収束する)速さはどうかを解析し、保証するためにも数学が必要になります。

このように、現象のモデル化、離散化、アルゴリズムの開発と解析、すべてにおいて数学(数理)が必要になります。つまり、計算機を使っていろんな現象をシミュレートすることを情報学の一つのミッションとするなら、そのあらゆる局面で数理が必要だということがおわかりいただけるでしょう。

佐藤 現象のモデル化と離散化、アルゴリズムの開発と解析——そのいずれも、実社会の現象を、いかに数学に置き換えて、わかるように提示するか、ということですね。

速水 そうですね。その使命は現実の世界にできるだけ近づくことにあり、より精度を高めるために、つねに、モデルも計算法も計算機も進化が求められているのです。

数学的な難問を解決する喜び

佐藤 それぞれ、具体的にはどのような研究に取り組んでいらっしゃるのですか？

龍田 私が手がけているのは、プログラム理論と数理論理学ですが、前者は先述のソフトウェア検証などのための数学理論、後者がギリシャ時代から



龍田 真
Makoto Tatsuta
情報学プリンシプル研究系教授



速水 謙
Ken Hayami
情報学プリンシプル研究系教授

Mathematics and Logic as the Cornerstone

続く学問といえます。最近の私の研究成果としては、プログラム理論の国際学会が未解決問題として挙げた 22 の問題のうち、20 番目の「全単射に相当するラムダ式を特徴づける型理論を与えよ」という問題を、2 年前に解決したことです(図1)。言い換えると、これは、「1対1対応を表す抽象的プログラムを、データの種類の分析で特徴づけよ」という問いですが、結果としては「それは存在しない」ということを証明しました。22問の未解決問題のうちで最初に解決した問題であり、ソフトウェア理論の進展に貢献するだけでなく、真理の探究という意味においても大きな成果といえます。

速水 私のほうは、大規模な連立一次方程式や最小二乗問題を計算機で解くための計算法の研究を行っています。大規模な連立一次方程式というのは、先述の流体の方程式を離散化した際に現れるものです。たとえば、空間を各方向から網目で 100 に切ると、空間全部で 100 万の接点に分解されます。となると、100 万元の連立方程式を立てることになる。それをいかに効率的に解くか、というのが課題です。

もう一つの最小二乗法とは、もともと 19 世紀の数学者ガウスが、天体の軌道の予測や測量のために開発した手法です(国立情報学研究所平成 19 年度市民講座「社会に生きる数学」参照)。応用と

してはほかに、実験などで生じる大規模なデータの統計解析、画像処理、信号処理、制御など多岐にわたります。

大規模な最小二乗問題を計算機で解こうとすると、繰り返し計算して解を改良していくという、「反復法」が重要になります。私たちは現在、大規模で解きにくい、たちの悪い問題でも高速に解けるような計算法を開発したり、その方法の振る舞いの数学的な解析を行ったりしています。大学院生が新しいアイデアを次々に考え出してきてくれるので、一緒に楽しみながら研究に取り組んでいるところです。

佐藤 いずれも一般の人にはほとんどなじみのない難しいテーマではありますが、科学技術の発展に、数理、理論が不可欠だということがよくわかりました。

数学の意義は好奇心と感動にある

佐藤 一方で、数学というのは、具体的に社会の役に立つことばかりではありません。とりわけ純粋数学では、真理の探究にこそ重きが置かれています。

龍田 数学の歴史を振り返ってみると、真理の探求として意義のある定理や法則は、数十年、あるいは数百年後に、結果として、実社会で役立ってきた例がいくつもあります。逆に言えば、最初から役に立てることを念頭に研究をしているわけではない、ということなんですね。

私は、数理、論理の意義には三つあると考えていますが、最も大切なのは好奇心だと考えます。人間が知りたいと思う気持ちに答えて、心を豊かにすることにこそ数学の意義がある。たとえば天文学も、あの美しい星のかなたはどうなっているのかを知りたい、という気持ちを原動力にして始まった学問です。要するに数理や論理は、文学と同様の意義をもつということです。

たとえば、幾多の天才数学者が長年にわたって挑み続けた「フェルマーの定理」($x^n + y^n = z^n$ となる整数 $n > 2$, $x, y, z > 0$ は存在しない)も、どのような地図も 4 色で塗り分けられるという「四色定理」も、あるいは、大きな素数についての疑問も、すべてのプログラムを模倣できるようなプログラム、いわゆる

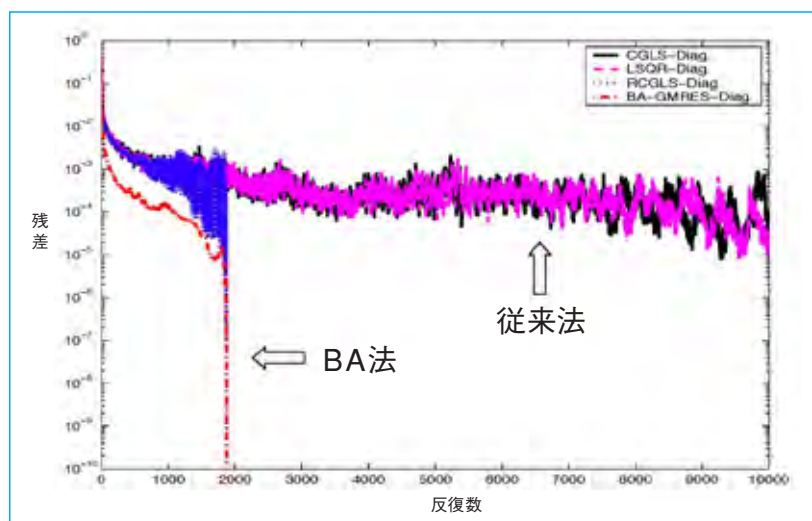


図2 BA法と従来法の比較
BA法という大規模な最小二乗問題に対する新しい反復解法を開発した。BA法を用いると、従来法に比べてはるかに少ない反復数で最小二乗解を求めることができる。

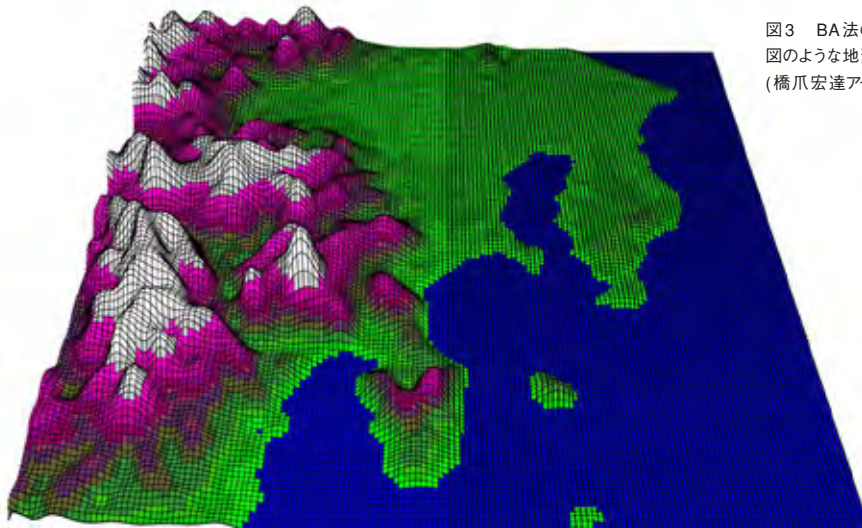


図3 BA法の最小二乗問題への適用例
図のような地形データの補間に用いることができる。
(橋爪宏達アーキテクチャ科学研究系教授作成)

る「万能チューリングマシン」は存在しない、という難問も、すべての源泉は好奇心にあります。また、文学や芸術は、人間が価値を感じて感動するから存在しているのだと思いますが、それは数学や論理も同じです。先述の難問の解決というのは、まさに感動的です。

佐藤 確かに、いい証明を見ると、いい仕事をしているなと思うし、美しいと感じますよね。それは、科学者や数学者がもつ共通の感覚であり、難問と呼ばれるものは、数学者が純粋にそこに面白みや感動を感じるから、全身全霊をかけて取り組むことができるのでしょね。

速水 興味深いことに、好奇心や感動を与えるような、数学者が直感的に美しいと感じるような理論であればあるほど、結果的に社会に役立つ場合が多いのも事実です。

龍田 ただ残念なことに、数学の意義として好奇心や感動というのは一般の人にはわかりにくいいため、現在は、社会の役に立つという意義が強調されすぎているくらいがあります。実際に、科学技術の発展に、数理、論理は不可欠に違いありません。ところが数理、論理の研究は非常に息が長いもので、フェルマーの定理が解かれたのは、360年後の1994年ですし、四色定理も120年後の1976年でした。さらにそれらが、具体的に役に立つかどうかわかるのは、ずっと先になってからのことが多いのです。ギリシャ時代から研究されている論理はいまやコンピュータに不可欠ですし、四色定理はいろいろな計算の最適化に、そして、大きな素数は暗号の分野で役立っていますが、そうなるには、数十年、数百年という時間が必要でした。

佐藤 数学に限らず、物理でも同様のことがいえ

ますね。ニュートンの万有引力も、我々の生活の何かに役立たせるつもりで発見されたものではありませんが、そこからアインシュタインの相対性理論が生まれ原子力エネルギーの開発に進んでいったと考えられます。

速水 もちろん、何かを解決したいという欲求から理論が生み出されることもあるわけですから、科学技術の発展に役立つということを否定的に捉える必要はないでしょう。

ちなみに、ガウスという人は、若いうちは数論の研究をしていたのですが、その後天文学へ転向し、最小二乗法を考案、彗星の出現を予言して一躍有名になりました。さらに最小二乗法を測量や地図の作成に用いています。また、測量の問題に取り組む中から微分幾何学のきっかけを創ります。この微分幾何学は後世、アインシュタインの一般相対性理論の構築に用いられることになります。このように学問というのは、理論が世に役立ち、さらに好奇心を刺激して、新たな理論を生むという連鎖構造をもっているんですね。

龍田 さらに付け加えるなら、数学者の真理に対する直感と良心だけが、数理、論理の正しい研究方向を与えることができるということではないでしょうか……。

佐藤 役に立つだけだと、問題が解決したらおしまいです。数学のような深い学問では、一般性があるので、さまざまな応用ができる可能性がある。それがまた新たな理論を生み出し、思いがけない分野で爆発的に役に立つということにつながるということでしょうね。

本日はお忙しいところありがとうございました。

(取材・構成 田井中麻都佳)