証明支援系を用いたトポロジーの形式化について

久我　健一　(千葉大学)*

近年、証明支援系を用いた「形式化」による定理の検証が相次いで報告されている。形式化によって、数学の定理は完全に検証されると同時に、その論理的な内容が電子データになると考えられる。現時点では形式化に要する労力はかなり大きく、形式化の完了した定理は多くない。されにシステム間の互換性の問題など、普及への障害は大きい。しかしこの状況は徐々に改善され、トポロジーも含めて、数学の様々な分野で形式化による電子データ化が進み、いずれ、これを基盤とするコンピュータの積極的活用が行われることは間違いないであろう。この発表では、主に幾何的トポロジーの例を用いて、始まったばかりの形式化の一端を見ていきたい。

1. 形式化と証明支援系

1.1. 形式化

「形式化」とは、定理の証明の各ステップで、どの推論規則や公理が使われたかを例外なく全て記録することである。これはコンピューターの登場以前からある考え方で、ヒルベルトが提唱したことで著名な公理論・形式主義が理想とする証明を行うことに他ならない。初期の形式化の試みとして、ラッセル・ホワイトヘッドの「数学の原理」(1910〜1913) が著名である。ここでは集合論、基数、順序数、実数が扱われている[1] このような、紙とペンによる「形式化」は多大の労力を必要とし、かつ正しく形式化されたかをチェックすることも難しい。従って、「形式化」が現実的に実行可能になるためには、一般的にはコンピューターの出現を待たなければならなかったと言ってよいと思われる。

コンピュータが形式化を行うようになれば、それは公理論、形式主義に則った証明の実行であるという意義の他に、証明をコンピュータで検証するという実際的な意味が出てくる。実際、定理の証明が非常に複雑であったり、非常に高度な難解なものである場合、査読にかけられる時間も限られ、また査読できる専門家も限られると、どうしてもルーチンのように見える部分は細部までチェックしないこともありうる。そこで、そのような定理を使おうとしたときは、自分で納得するまで証明を読むしかない。しかしそれはとても時間がかかるし、さらに自分自身が本当に信用できるか不安は残る。その点、コンピュータによる検証は（ソフトウェアにバグがあるとか、仮定のおき方にミスがあるというようなことを除いて）完璧と考えていいだろう。その証明がコンピュータによって検証されていたら、定理の内容を理解し、証明のポイントに納得したら、安心してそれを使って先に進むことができるのではないだろうか。ホモトピー型理論の提唱者の一人であるボエボドスキーの動機の１つも、そのような点にあ

本研究は科研費 (課題番号:15K13433) の助成を受けたものである。
キーワード：幾何的トポロジー, 証明支援系, 形式化
* 〒 263-8522 千葉市稲毛区弥生町 1-33　千葉大学 大学院理学研究科
e-mail: kuga@math.s.chiba-u.ac.jp
[1] もっともこの大作は誰も読んだことがない大著という話もある。… ゲーデルを除いて … というお話もある。ついでながら、ゲーデルの不完全性定理によってヒルベルトプログラムは破綻した筈ではないか、と疑問を感じ始められた方のために。その意味では、この話はゲンツェンの無矛盾性定理から後述のカリー・ハワード対応に発する流れであって、不完全性定理と直接的な関係はない。たとえば [11] 参照

るらしい。

さらに進んで、コンピュータが数学証明の検証をすることができるなら、コンピュータ自身で数学定理の証明をすることもできるようになるのではないか、と考えることも、自然であろう。自動検証から自動証明は現時点ではかなりの飛躍があるかもしれないが、簡単な自動証明機能であれば、現在の証明支援系で使えるものもあり、将来的には機械学習などを利用して、コンピュータの自動証明機能が向上していくことは間違いないと思われる。そうなると、形式化された数学は、コンピュータが直接操作対象とできるデータという意味で、ますます重要になって行くと考えられる。実際、インターネット上に電子化された数学の書籍や論文は多数あるが、それはほとんどpdfファイルであり、コンピュータがそこから数学の論理的内容を読み取れるわけではない。しかし、それらがコンピュータが直接理解でき、操作することのできる形式言語で書かれていたら全く話が違ってくる。

1.2. 証明支援系

コンピュータを用いて「形式化」を行い、その検証を行うためのソフトウェアが証明支援系 (proof assistant) である。証明支援系は通常

- 数学の定義や定理、証明を記述するための形式的言語
- ユーザーと対話的に証明を作成するための環境

を提供する。従って、「数学証明の自動検証」のためのシステムであり、「数学の定理の自動証明」システムではない。しかし、数学の定理の自動証明がこのようなシステムを前提とすることは間違いない。たとえば、後述のCOQ等には、若干の自動証明機能がある。

このようなシステムの初期のものとしてAutoMath　(Nicolaas Govert de Bruijn starting in 1967) がある。この段階ですでにカリー・ハワード対応として知られる考え方や、依存型系（dependent type system) が取り入れられていた。

Mizar(http://www.mizar.org/) というシステムは現在も開発が続けられている。現在良く知られている証明支援系としてはHOL, Isabelle, COQ, Agda 等がある。このうちCOQとその拡張であるSSReflectについて後で若干詳しく述べる。

近年このような証明支援システムによって多くの定理の形式化が達成されている。幾つか例を挙げると

- 4-color theorem (Coq/Gonthier/2004)
- Jordan curve theorem (Mizar/Kornilovicz/2005, HOL light /Hales/2007)
- prime number theorem (Isabelle/Avigad et al/2007)
- Feit-Thompson theorem (Coq/Gonthier/2012)
- Kepler conjecture (HOL, /Hales/2014)

定理の証明が複雑になると、このような証明支援系を用いても、労力は大変なものとなる。2014年8月に完了したKepler conjectureの形式化は280[年 × 人] を要したと

いう計算もある。しかし、このような労力も証明支援系の自動証明機能の向上で緩和されていくと思われる。実際後で少し触れるCOQのSSReflcet拡張は自動証明機能が強化されていて、SSReflect を用いた形式化のコード量はかなり少なくなる傾向がある。

また、現時点でグラフ理論、有限群論、線形代数、組合せ論に属する問題は形式化が進んでいる。これはたとえばSSReflectのライブラリがそのような分野をカバーしているということでもある。しかしトポロジーでいうと、特異ホモロジー論からすぐ従うジョルダンの閉曲線定理の形式化が2005年であることからも分かるように、基本的な代数トポロジーも整備されていない。もちろん、位相空間の公理を形式化すること自体は簡単で、位相空間のライブラリはいくつかあり、我々は、それを用いて幾何的トポロジーの基本的な定理を形式化できることを後で示す。しかし、そもそも形式化で点集合としての空間を基本的な対象として扱うのは適切なやり方ではないかもしれない。これについては、後で少し触れるHomotopy Type Theoryとも関係している。

2. 型理論とカリー・ハワード同形対応

型理論はもともとB. ラッセルが提唱したもので、素朴集合論のパラドックスを回避するのが目的であった。公理的集合論も、このパラドックスを回避するもう一つの方法であり、現代数学は、いわゆるZFC理論を基礎にしていると考えられる。しかし、Ｃｏｑのような証明支援系では型理論を使用する。(さらに、型理論によって数学を基礎づけるという立場もありうる)。多くの数学者にとって、型理論は馴染みがないと思う。特に「命題は型である(propositions as types)」とか「証明はプログラムである」というカリー・ハワード同形対応は、集合論に基づいた通常の数学で命題と集合が全く別のものであるのと対照的で、全く新鮮な考え方ではないかと思う。そこで、少し例で説明をする。例えばC OQで定理は次のように書く：

```
Variable p : Prop.
Theorem p_implies_p : p → p.
```

一行目は p は　Ｐｒｏｐ型の項という意味である。　つまり p は Prop という集合のようなものの元であるという意味である。

二行目は p → p という主張に　p_implies_p という名前を付けている。

しかし、二行目は一行目と同じで p_implies_p は p → p という命題型の元であるという意味でもある。これはまた、p_implies_p は　p → p　の証明であるとか、証拠であるという意味である。

このように、命題を、その証明全体のなす型と考えることで、命題と数学の対象とを同じものとして扱うことが可能になっている。

さらに、このことは、数学の対象を帰納法などで構成するのと同じように、p_implies_pを構成すれば、それが定理の証明をすることになる。こうして数学の対象を構成するように証明を行うことができる。このような構成的型理論はMartin-Löf型理論と呼ばれる。

3. COQ/SSReflect

今例に使用した COQ はフランス国立情報学自動制御研究所が中心となって開発をしている証明支援系である。

たとえば自然数は次のように定義される：

```
Inductive nat : Set :=
```

```
  | O : nat
  | S : nat → nat.
```

このような帰納的構成が基本的である。

自然数を書いたので、ついでにCOQでの実数の定義も解析学の教科書に書いてあるように公理的に行い、通常のプログラミング言語の実数が浮動小数点であるのとは全く違っていることいも注意する。実数の公理的な定義の仕方はいろいろあると思うが、たとえばCOQの標準ライブラリの実数の公理の一部は完備性の仮定は次のようになっている：Raxioms.v

```
Axiom
  completeness :
    ∀ E:R → Prop,
      bound E → (∃ x : R, E x) → { m:R | is_lub E m }.
```

先に書いたように、COQを用いてG.Gonthierのチームが４色定理の形式化や、ファイト・トンプソンの定理の形式化を達成している。またSSReflectという拡張がGonthierのチームによって開発されており、保守性や、自動証明機能が高められている。Mathcompという（有限集合や代数の）ライブラリが開発され、これらのライブラリを使用することによって、短いコードで形式化を行うことが可能になっている。

我々はこのCOQ/SSReflectを用いて、次に述べるように、幾何的トポロジーでいくつかの形式化を行った。

4. 幾何的トポロジーの形式化

前述のSSReflectの数学ライブラリーMathcomp等は、まだトポロジーや解析学をほとんどカバーしていない。もちろん、公理を書くことは簡単であって、たとえばCOQを用いて位相空間の公理は次のように書ける[2]：

```
 Record TopologicalSpace : Type := {
  point_set : Type;
  open : Ensemble point_set → Prop;
  open_family_union : ∀ F : Family point_set,
    (∀ S : Ensemble point_set, In F S → open S) →
    open (FamilyUnion F);
  open_intersection2: ∀ U V:Ensemble point_set,
    open U → open V → open (Intersection U V);
  open_full : open (@Full_set _)
}.
```

位相空間のライブラリもいくつか作られているが、標準的なものといえるものは出来ていない。実際、上記のように位相空間を形式化しても、部分位相空間や誘導位相などを考え始めると、様々な面で困難が発生する。たとえば、部分集合は通常命題で指定されるが、ある点がその部分集合に入っていることは、その命題の証明があって、初めて確定する。トポロジーでよく使われるフィルトレーションのようなものを考えると、この証明の扱いが非常に面倒になってくる。また、距離空間からくる部分空間の位相と、相対位相が同じものであることが、決して自明には使えない。従って、位相空間の使いやすい標準的なライブラリの開発が望ましいが、それはまだ出来ていない。

[2]通常、空集合が開集合であることは公理に含めるが、定理として得られるので省いてある。

我々が形式化したのはビング収縮定理とよばれるビングトポロジーの基本定理で若干テクニカルになるが次のようなコードになる。

```
Hypothesis X_compact: compact Xt.
Hypothesis Y_compact: compact Yt.
Definition Bing_shrinkable (f:X→Y): Prop:=
  ∀ eps:R, eps>0 →
    ∃ h : point_set Xt → point_set Xt,
     homeomorϕsm h
    (∀ x:X, d' (f x) (f (h x)) < eps)
    (∀ x1 x2:X,  (f x1) = (f x2) → d (h x1) (h x2)  < eps).

Definition approximable_by_homeos (f:X→Y): Prop:=
  ∀ eps:R, eps>0 →
    ∃ h:point_set Xt → point_set Yt,
     homeomorϕsm h
    (∀ x:X, d' (f x) (h x) < eps).

Theorem Bing_Shrinking_Theorem:
 ∀ f: point_set Xt → point_set Yt,
continuous f → surjective f →
 (Bing_shrinkable f → approximable_by_homeos f).
```

この証明の形式化は github [9] にある。そこには、このために必要なベールカテゴリー定理の形式化も行われている[3]

```
Variable T: Topological_space.
Definition baire_space : Prop :=
 ∀ V : IndexedFamily nat (point_set T),
  (∀ n: nat, (open (V n)) ∧ (dense (V n))) →
   dense (IndexedIntersection V).

Theorem BaireCategoryTheorem :
                  complete d d_metric → baire_space.
```

この定理は基本的には選出公理であり、選出公理を仮定していることは断っておく必要があるとおもう：

```
Axiom FDC : FunctionalDependentChoice_on
              (point_set X * { r:R | r > 0} * nat).
```

ビング収縮定理は、もともとは、R.H.Bing が３次元球面の位相的 involution で固定点集合が標準的でないものを構成する際に用いた [1]。この定理は幾何的トポロジーで同相写像を構成するときの基本的な方法を与える定理であり、考え方としてはモートンブラウンの一般シェンフリース定理 [2] と同様である。

我々はシェンフリース定理をさらに一般化する M. Freedman による次の定理の形式化を目指している。

Theorem (Disk to Disk Theorem for $k = \infty$)　Suppose $f : S^n \to S^n$ is a surjective map such that there are only countably many points p_i, $(i \in \mathbb{N})$ with

[3] 但し、ベールカテゴリー定理については形式化はすでに成されており、最初の形式化ではないことが後からわかった

the property $\mathrm{Card}(f^{-1}(p_i) > 1$. We assume $\lim_{i\to\infty} \mathrm{diam}(f^{-1}(p_i) = 0$ and the subset $\{p_i | i \in \mathbb{N}\} \subset S^n$ is nowhere dense. Then f is Bing shrinkable.

この定理が重要なのは、Casson Handleが位相的には2-ハンドルであるというFreedmanによる４次元多様体の分類の基礎にある定理の主要な補助定理であることである。

さて、たとえばFreedmanの一般シェンフリース定理を証明しようとするとき、円板D^nや球面S^{n-1}はどのように形式化すべきか考えると、すぐに思いつく

$$\begin{aligned} D^n &= \{(x_1, x_2, \ldots, x_n) \in \mathbb{R}^n \mid x_1^2 + x_2^2 + \cdots + x_n^2 \leqq 1\} \\ S^{n-1} &= \{(x_1, x_2, \ldots, x_n) \in \mathbb{R}^n \mid x_1^2 + x_2^2 + \cdots + x_n^2 = 1\} \end{aligned}$$

は具体的なため、逆に汎用性がなくなり、形式化も煩雑になる欠点がある。たとえば、円板内部の任意の２点が境界を固定する円板の同相写像で移り合うことを示そうとすれば、円板よりも立方体のほうが使いやすい。そこで円板と立方体の同相写像を作る必要も出てくる。従って円板を初めから具体的な集合として定義するよりも、境界を固定した同相写像の性質などの抽象的な性質をもった空間を一つの型としておくほうが、よほど形式化しやすく、かつ汎用性も高まる。これはすべての点で言えることで、いっそのこと、空間が点からできているという解析幾何的な方法はやめて、統合的幾何にしようという考えは、このような形式化をやってみると自然な発想として出てくる。これはちょうど、抽象的なホモトピー論の考え方と同じで、次に少し述べるホモトピー型理論のような数学の基礎づけが現れるのも必然かもしれない気がしてくる。

5. ホモトピー型理論

最初に断っておく必要があるが、発表者はホモトピー型理論をまだよく理解していないので、ここに書くことは、思い違いなどがあるかもしれない。しかしここ２〜３年でプリンストンで精力的に行われている抽象的ホモトピー論を手本にした、数学の新しい基礎づけについて少しだけでも触れておきたいと思う。本もネットでダウンロードできるので、正確にはそれを読んで頂ければと思う。

http://homotopytypetheory.org/book/

またボエボドスキーの講演も

https://video.ias.edu/univalent/voevodsky など、幾つかネットで視聴することができる。

ボエボドスキーの動機は、専門的な難しい論文になればなるほど、証明に欠陥があっても、発見することが難しくなっており、そのような論文の結果の信頼の上になされなければならない研究にとって、深刻な問題であり、自動的な証明の検証システムが必要であるということである。そこで証明検証系を利用してみたが、数学者にとっては、本格的に使うにはいろいろ問題がある。

ひとつは、数学者が当たり前のように行う同一視がコンピュータ上では簡単でないという点である。簡単な例では、公式$\displaystyle\sum_{k=1}^{n} k = \frac{n(n+1)}{2}$は、$k$や$n$が自然数であろうと整数であろうと、どちらにせよ、証明も含めて、まったく同じく扱える。しかし、自然数か整数かによって二つの公式（命題）はまったく別の型になり、そのような同一視は自動的ではない。

また、上に書いた球面の例でも、数学者が球面を考える時、それは必ずしもユークリッド空間の部分集合を考えているわけではなく、時と場合に応じて、ある抽象的な

性質をもつ型としての空間を想定しているという点である。このようなことは、数学者が頭の中で無意識に処理している内は深刻な問題にはならなかったが、全く融通の利かないコンピュータによる形式言語では、その無意識の処理を明示的に定式化する必要が出てくる。

そこで、ホモトピー型理論が提唱する数学の”univalent foundation”では、抽象的なホモトピー論にその手本を見つけようとうしている。たとえば型の属（ある型でパラメータづけられた型）はファイブレーションであるなど。

HoTTのライブラリでは球面がユークリッド空間の部分集合とか多様体ではなく、基本的な型として帰納的に定義されている部分を載せておくと hit.Spheres.v

```
Fixpoint Spheres (n : trunc_index)
  := match n return Type with
    | -2 ⇒ Empty
    | -1 ⇒ Emply
    | n'+1 ⇒ Susp (Sphere n')
  end.
```

ちょうど自然数の定義のようになっていることがわかると思う。このように帰納的構成もうまく噛み合っている。

参考文献

[1] Bing, R.H.: A homeomorphism between the 3-sphere and the sum of two solid horned spheres, Ann. Math. 56, 354–362 (1952)

[2] Brown, M.: A proof of the generalized Schoenflies theorem, Bull. Amer. Math. Soc. 66, 74–76 (1960)

[3] Casson, A.: Lectures on new infinite constructions in 4-dimensional manifolds, À la recherche de la topologie perdue, Progr. Math., vol. 62, Birkhauser Boston, Boston, MA, 1986, pp. 201–244. With an appendix by L. Siebenmann.

[4] Freedman, M.H.: The topology of four-dimensional manifolds, J. Differential Geom. 17, no. 3, 357–453 (1982)

[5] Freedman, M.H.: Bing topology and Casson handles, notes by S. Behrens, 2013 Santa Barbara Lectures

[6] Freedman, M.H., Quinn, F.: Topology of 4-manifolds, Princeton Mathematical Series, vol. 39, Princetion University Press, Princeton NJ (1990)

[7] Hales, T.: The Jordan curve theorem, formally and informally, The American Mathematical Monthly 114(10), 882–894 (2007)

[8] Hales, T. et. al :A formal proof of the Kepler conjecture, http://arxiv.org/pdf/1501.02155.pdf

[9] Kuga, K.: Bing_Shrinking_Criterion, `https://github.com/kenkuga/Bing_Shrinking_Criterion`

[10] Schepler, D.: Topology/v8.4 `http://www.lix.polytechnique.fr/coq/pylons/contribs/view/Topology/v8.4`

[11] 照井一成 「コンピュータは数学者になれるのか？　数学基礎論から証明とプログラムの理論へ」 青土社 (2015)