数学定理の証明のコンピュータ（証明支援器）による形式化の研究

渡瀬泰成 †

† 立正大学

†ywatase@ss.iij4u.or.jp

キーワード　形式証明, 形式化数学, 機械証明, Mizar

1　はじめに

数学に関連した分野でのコンピュータ利用としては数値解析での高水準コンピュータ言語による数値計算, 代数記号の計算が出来る計算機代数（ネター環上の多項式環のイデアルの基底計算）ソフト Maple, Risa/Asi, Magma 等々, 技術系論文向けの DTP ソフト Tex, LaTex があり最早や紙と鉛筆があれば数学研究をするスタイルとは異なる別のスタイルが形成されつつある．本稿では種々ある数学に関わるコンピュータソフトの中でも証明に関連するものを扱い, 証明支援器[1]と呼ばれる証明を形式化し検証するソフトを用いての研究について紹介する．

2　証明の形式化とは

数理論理学に於いて 1 階述語論理で表される命題の証明は推論のステップごとに証明図[2]で記述される．これを形式化された証明という．電子計算機の登場に伴い 1960 年代に米国を中心に計算機の定理証明への応用が研究され Daivis や Wang 等により定理自動証明が研究された．70 年代後半にポーランド、ワルシャワ大学のグループが 1 階述語論理による証明支援器 Mizar を開発, 実装した．80 年代になると型理論が整備され、型理論に基づいた高階述語論理による証明支援器 HOL や Coq が出現した．数学証明以外にプログラム・アルゴリズムの検証や集積回路の検証に応用され研究されている．数学理論の証明について言えば, 場合分けが膨大な証明、四色問題, Feith-Thompson の定理（奇数位数の群は可解）の定理の検証[3], 日本での Mizar 研究拠点である信州大学の研究グループも大いに貢献してジョルダンの閉曲線の定理が検証[4]された．

2.1　形式化証明と証明図の外観

証明支援器の処理を支える論理の枠組みとして 1 階等式述語論理を採用するもの (Mizar) と高階述語論理を採用するもの (Coq, HOL, Lean 等) に分かる．定理証明の記述は公理と既に証明されている定理から有限回の推論で目的の定理に至る推論過程を記述したものと定義される．“Formalizing 100 Theorems”[7] と題した Web ページに提示されおり, お題として 100 の定理が与えられそれらに対する 10 種の証明支援器 (HOL Light, Isabelle, Metamath, Coq, Mizar, Lean, ProofPower, PVS, nqthm/ACL2, NuPRL/MetaPRL) による形式化の結果が一覧できる．各々の証明支援器による定理とその証明の具体的記述の様子が分かる．Mizar について言えば, 証明の形式記述は人の手により作成される．HOL や Coq では対話式に証明したい論理式から出発し推論を支援する機能を用いて推論過程を記述する．冒頭に述べた証明図であるが, 正確に説明するには 1 階述語論理の準備を要するが簡単に証明図を見ておく．

$$\dfrac{\dfrac{\dfrac{[P\wedge(P\to Q)]}{P\to Q}(\wedge E)\ \dfrac{[P\wedge(P\to Q)]}{P}(\wedge E)}{Q}(\to E)}{P\wedge(P\to Q)\to Q}(\to I) \quad (1)$$

この証明図は命題論理のものであるが三段論法の証明図となる．図の読み方は横棒の上が仮定であり下が結論となる．

$$\dfrac{[A\wedge(B\to C)]}{B\to C}(\to E) \quad (2)$$

(2) の図は, A が成り立つ, 且 $B\to C$ が成り立てば, $B\to C$ が成り立つと読む．$(\to E)$ は $(A\wedge A\to B)$ から B が成り立ち $\to$ を消去する推論規則であり, $(\wedge E)$ は $(A\wedge B\to C)$ から A を消去しても棒の下の命題が成り立つという推論規則を示している．E でなく I となると論理式の挿入と読み替え命題が挿入できる場合の推論規則を示す時に用いる．(1) の図の一番上に, (2) の形の証明図が二つ

[1]定理証明支援器とも証明支援系とも言われる．本文では短く証明支援器と呼ぶ．

[2]証明図は証明木とも呼ばれる．

[3]Coq により 2012 年に形式化が完了, 検証された．

[4]Mizar により 2005 年に形式化が完了, 検証された．

並ぶが, これは 2 つの図式が両方成り立っている前提と読む．以上のような仕組みで証明は構成される．命題が推論規則と理論の構成の出発点で決めた公理を使って有限の長さの証明図があるとき命題は証明を持つ[5]という．

3 研究内容

3.1 証明の形式化の研究意義

重要な数学結果には徹底した形式化による証明を付けて電子的に保存するシステムの構築を謳った 1994 年の QED マニュフェスト [4] に向けた学術的な活動と見做すことも可能である．これを数学定理証明と呼ぶことにする．しかしながら現在の証明支援器の進歩と PC 性能の向上によりシステム, プログラムの正当性, 認証や通信プロトコルの正当性の検証への応用, システムの代数的仕様記述に応用されシステムの信頼性という観点で重要な分野となり工学的応用に寄与している．筆者が主に形式化に利用している Mizar による形式化は, 主に可換環の定理群と整数論の定理の証明の形式化研究であり数学定理証明の分野への貢献を目指すものである．

3.2 Mizar の発達経緯

数理論理学に於いて 1 階述語論理で表される命題の証明は推論のステップごとに証明図で記述される．これを形式化された証明という．証明が長い数学の定理を形式化した証明図を紙面上で展開し記述することはスペースをとり困難である．証明の記述の工夫が古くから考案されてきた．1920 年代のポーランド表記を発案したウカシェヴィツェ（Jan ukasiewicz,1878-1956）の先行研究 l があり、コンピュータの発達ともに 60 年代よりコンピュータを利用した研究がなされ証明をコード化し証明の推論過程を検証するソフトや命題の自動証明が考案されてきた．日本にあっては名古屋大学の小野勝次が実際的な証明記述 [1] を発案した．70 年代にポーランド, ワルシャワ大学のグループが 1 階述語論理, Tarski-Grotandieque 集合論をベースとした形式化した証明の機械検証を実現するシステムとして Mizar を開発実装した．当初の目標は数学の定理を見やすい表現でコード化しその証明も機械的に検証されるドキュメントを電子的に作成する、検証された定理はデータベースに蓄積され再利用できる様にすることであった．証明記述法と推論規則は同様のシステムが前述 [1] に於いてすでに考案されていたと開発・発達の経過をまとめた論文 [3] において指摘している．1980 年代に形式化された証明を Tex に変換し通常の数学記述に近い論文を展開する機能が開発された．1990 年に創刊された Formalized Mathematics 誌の論文はこの機能を活用して出版されている．例えば集合論の一連の部分集合に関する公式群は識別子.miz を持つ SUBSET1.miz と名前のファイル記述されている．ファイルはアーティクルとも呼ばれる．論文投稿のプロセスを大まかに言えば, 新規に何かしらの定理の証明を形式化し投稿する場合は, コード化された形式化した証明をその内容を説明したアブストラクトを添え投稿する．コードの内容は複数人の査読者により査読され証明自体は Mizar により機械的に検証される．受理された証明コードは Tex への変換により得られるドキュメントを論文として論文誌に掲載される．この様に形式化された数学の定理の蓄積が遂行されている．ここで Mizar での証明の例を簡単に見ておく, R を 2 項関係として,「R が対象律と推移律を満たすならば反射律を満たす.」なる定理, これは [2] で使われた例であり小野が提示した証明図と対比し易いように行のラベルもあわてある．比較すると大変近い表記であることが分かる．論理記号で表現すると

$$\forall x, y(xRy \wedge yRx) \wedge \forall x, y, z(xRy \wedge yRz) \rightarrow \forall x, y(xRy \rightarrow xRx)$$

となり, Mizar でのコード化では以下の様に記述される．

```
theorem Example1:
   (for x,y st [x,y] in R holds  [y,x] in R &
   for x,y,z st [x,y] in R & [y,z] in R
   holds [x,z] in R) implies
   for x,y st [x,y] in R holds [x,x] in R
   proof
     assume
A:   for x,y st [x,y] in R holds  [y,x] in R &
     for x,y,z st [x,y] in R & [y,z] in R
     holds [x,z] in R ;
b:   for x,y st [x,y] in R holds [x,x] in R
     proof
       let u,v;
       assume
bA:    [u,v] in R;
       reconsider x=u as Element of field R;
       reconsider y=v as Element of field R;
bb:    [u,v] in R implies [v,u] in R by A;
       [x,y] = [u,v] ;
bc:    [v,u] in R by bA,bb;
bd:    [u,v] in R implies
       ( [v,u] in R implies [u,u] in R) by A;
```

[5] provable とも言う.

```
be1:   [u,u] in R by bA,bc,bd;
        hence thesis;
      end;
      hence thesis;
    end;
```

図 1. Mizar コード例

このコーディング例では「proof〜end」の中に証明が記述される．ラベル A:で前提を記述している．ラベル b:で証明すべき論理式が続く．推論過程ではラベル bc:の内容は, ラベル bA:とラベル bb:から推論されると読み「by bA,bb」と記述する．by に続くのはラベル名, 既に証明された定理の名前[6], 定義の名前などが続くのである．詳しい解説は [8] を参照されたい．

3.3 論理の枠組と数学表現力

証明支援器に実装される論理に 1 階述語論理を取るか高階述語論理を取るかにより表現可能な数学記述に差が生じる．1 階の述語論理であると量化子により限定される変数は領域 M の元のみであり, 2 階の述語論理では変数が M で定義される関数全体わたることや M の部分集合をわたることが扱える． 排中律については 1 階論理では成立するが, 2 階では成立しない．例えば可換環 A の積閉集合 S による分数環 $S^{-1}A$ の普遍性は可換環の圏の言葉で次の様に表せる．$(\forall f)(f(S) \subset \{cod\ f \text{ の単元}\} \to \exists g(f = g \circ \pi))$

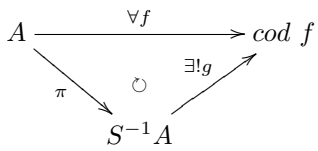

図 2. 可換図

ここで π は A から $S^{-1}A$ への標準射, f は値域が A なる射全体にわたる．f は写像であり変数なので 2 階論理の記述である．しかし任意に固定したと考えれば定数であり 1 階論理に落とし込む工夫が出来る [5]．弱い 2 階論理と云うべき表現が可能なのである.[7] このような表現を代数学の一分野である可換環の標準的な教科書 [6] の証明を底本として形式化を通じて表現技法を蓄積検討する．効率的な表現法を確立し理論的な裏付けを目指し研究している．

[6]他のアーティクル (仮に ARTICLE1.miz とする) の定理 (Th10) を参照する場合は ARTICLE1:Th10 の様に記述して参照する．

[7]Mizar では圏の概念も形式化されているが扱いやすい形式ではない．実際図 2. に現れる $\exists!g$ の構成は集合上の写像を構成しそれが一意でかつ環の準同形であることを示していくステップを踏む．

3.4 証明のコード化の実際

先ずはコード化に必要な Mizar で使用している言語を [8] 等を参考[8]に習得し簡単な題材のコード化を練習するなりして言語習得が必要であろう．数学書籍に記載される定理と証明は, 形式化されていないで読者層の知識を仮定して細かな議論や繰り返しの議論は省略する．これをインフォーマルな記述と言うことにするばインフォーマルな証明をコード化して形式化する場合, 定理の記述に必要な定義類を準備, 証明に必要な補題を定理同様に準備し形式化し定理本体の証明に加えすべて形式化する．既に形式化されている定理を前提条件して推論を進めることもできる．一概に工数は見積れない, 乱暴ではあるが経験的には文献 [6] で 1 から 4 章の中で形式化できる内容であれば 1 ページ 2 千〜3 千行のコードを要すると見当であるが見積れる．新規に概念を導入する場合は必要な定義や補題を準備する．証明したい定理の証明の推論過程で準備した定義と補題が整合しているか試行錯誤し数回書き直すと万単位の行を費やすことにもなる．また数理論理学, 特に一階述語論の知識も証明の推論を分析するのに重要となる．プログラマーの能力, 論理学の基礎, 数学の理解が入用となり, コーディングでは集中力と体力が必要となると苦労も多いと思われるやも知れないが, 証明の細部がすべて詳らかになり証明がすべて理解され報われる．数学書に書かれたインフォーマルな証明でも丁寧なもの, 論理推論が辿り易いようよく配慮された物など著作者の意図も理解出来るようにもなる．基本的な定理の証明の蓄積は基本的な課題である．形式化された定理のデータベースが豊かであればあるほど応用範囲が広がるからである．コード化の作業軽減の為の AI により類似の証明を探索するなどの創意工夫も近年考案/研究されている．これは自動証明機能がない定理検証システムの次に進む新たなフェーズの一つと思われる．他方基本的な定理の証明の蓄積が豊富であればその分様々な応用が可能とのるので地道な形式化証明の蓄積も重要な研究と言える．

4 おわりに

最後に証明の形式化の研究が重要かと言う根本的な問に対し若干の考察を与える．数理科学やそれに付随したソフトウエアは多様にある．現実社会の生活も多かれ少なかれ多様なソフトウエアに依存している．そこで動く

[8]英文であれば Mizar Home Page (http://mizar.org/) の”Bibliography of Mizar Project”の項目の中に様々なリソースのリンクが収録されている．

プログラムが正しく動くためにも形式検証は重要である．

形式検証を支える基礎技術が証明形式化の技術と考える．それ故に研究の重要性がある．この構図は現代数学にあっても存在している．高度な理論を構築して優れた証明に誤りがあっても簡単に判明しない状況もあり得るのである．ミルナー予想を解決したヴォエヴォドスキーは自らの講義の中で自身の論文の誤りを 20 年後発見した経験を語り, コンピュータを利用して証明の検証が出来るように新たな証明論の枠組みを提唱した．残念ながら同氏は 2017 年に没したが, ホモトピー型理論の研究は続いており証明の形式化は続くと考える．厳密な推論の検証を要する数学定理, プログラムに潜む誤りを発見する手段としての証明支援器は強力な道具立てである．証明支援器は様々なシステムのより高い信頼性, 安全性を担保する為の方法論を提供する能力を備えているのである．

参考文献

[1] Katuzi Ono. On a practical way of describing formal deductions. Nagoya mathematical journal, vol. 21. pp. 115121 (1962).

[2] 小野 勝次, 名古屋グループの論理学研究, 数学, 1968, 20 巻, 3 号, p. 154-164

[3] P. Rudnicki : An Overview of the MIZAR Project, Proceedings of the 1992 Workshop on Types for Proofs and Programs. (Bstad) (Bstad). pp. 291-310 (1992)

[4] Robert S. Boyer. Panel Discussion: A Mechanically Proof-Checked Encyclopedia of Mathematics: Should We Build One? Can We? - CADE 12, Springer-Verlag, Lecture Notes in Artificial Intelligence, Vol. 814, pp. 238-251, (1994).

[5] Yasushige Watase. Rings of Fractions and Localization. FORMALIZED MATHEMATICS 28(1) pp. 79 - 87 (2020).

[6] M.F.Atiyah, I.G.MacDonald. *Introduction to commutative algebra*, Addison Wesley Publishing Company, 1969

[7] Freek Wiedijk. Formalizing 100 Theorems, https://www.cs.ru.nl/ freek/100/.

[8] 中村八束，渡辺稔彦，田中保史，カワモト・ポーリン Mizar 講義録, 信州大学工学部　情報基礎研究室 http://markun.cs.shinshu-u.ac.jp/kiso/projects/proofchecker/mizar/Mizar4/index-j.html