

落書帳からの証明

7 N-4

— 計算機による論証支援のための証明方法論の一考察 —

南 俊朗, 沢村 一
(富士通㈱・国際情報社会科学研究所)

1. はじめに

数学の理論を作る場合やプログラムの正当性を証明する場合などの様々な場面で対象を論理的に記述し、その性質を明らかにし、問題の解決を図るやり方(「論証」と呼ぶ)が用いられる。人間の行う論証の過程を計算機で支援するための道具として、証明チェッカや証明コンストラクタが考えられている。(例えば[1]～[5])しかし、これらのシステムは、ある特定の論理系を想定し、その論理系における証明の検証、構築の支援を意図しているが、対象の表現のための論理系を1つに固定すると、問題によってはその論理系で表現することは不適切であることが起る。しかしながら、必要となる論理系毎に異なる支援システムを別々に作成するというのでは大変な手間がかかり、実際のやり方ではない。そのため、論理系そのものを定義し、記述された論理系に応じた論証の支援を行う「汎用の論証支援システム」を考察している。

そのような目的のシステムを実現するためには：①論理系を記述し、それを登録する機能、②与えられた論理系における証明の構築を容易にするための支援機能、③できあがった証明を整理し、格納する機能、が必要となる。本稿では証明支援機能を中心に論証支援に関する考察を行う。

以下、第2節では、論証の過程はどのようなものであるか考える。第3節では、我々が落書帳と名付けた、証明構築支援環境について説明する。第4節では証明の整理機能について述べる。

2. 論証過程

論証が行われる過程を次の3つに分けて考える：

①論理系の定義： まず、問題となっている対象を記述するために必要な論理的記述力を考慮し、それが表現できる論理系を定める。たとえば、時間的な関係を表現するためにはある種の時間論理が必要であろう。必要に応じて最適な論理系を既存のものの中から選択するか自作する。

論理系の記述は、論理式の構文定義、推論規則の構文定義、及び公理の設定からなる。即ち、論理式の中でどのような記号をどのような構文で用いるのか、をはっきりさせ、それらに対する基本的性質を推論規則および公理の形で

表現する。これらの性質は、その論理系を用いる限り共通に用いられるものである。

②証明： 設定された論理系の枠組みの中で、様々な状況を設定し、それらに関する証明を行い、考察したい論理的モデルの持つ性質を明らかにする。仮定を設けたり、公理を引用したり、以前既に証明した結果を探索したりしながら証明を行っていく。また、証明は仮定や既に得られた結論などを組み合わせて新しい結論を導くという、いわゆるボトムアップ的な接近法、逆に目的とする式をそれを導くための十分条件に分解する、トップダウン的な方法、役に立ちそうな中間結果を出しておく方法など多くの方法が存在するものである。

③証明の整理： 前ステップで試行錯誤的に構成された証明を整理し、まとまった定理の形で格納し、後の証明で用いることができるように備えておく。

論証の過程は、初めから完全に行われるものではなく、試行錯誤的に行われるものである。例えば、論理系に関して言えば、問題の表現に適した論理系は始めから厳密に定義されるものとは限らないので、②、③のステップへ進んだ後からでも必要に応じて、いつ①に戻るかもしれない。

3. 証明のための落書帳

論証支援システムはこのような試行錯誤的な論理系、証明の構築活動を支援しなければならない。そのための配慮が必要である。落書帳と呼んでいるのは正にこのような書いては、消し、加えては修正をするといった、いわばメモ用紙を使うような動作を支援するための構築環境である。

証明のための落書帳の備えるべき機能は次のようなものである：

- (i) 落書帳の上には公理、推論規則、既に証明された定理、仮定、を表現できる。これらは証明を作り上げるための基本要素となるものである。
- (ii) 落書帳の上で(i)の要素を組み合わせて、推論の段つきのステップを1つの要素として表現できる。また、新たに作った要素も組み合わせのための材料として用いることができる。
- (iii) 落書帳の上で一度構成した組み合わせを解除するこ

Proof Construction with Working Sheets

— A Consideration on the Methodology for Computer Assisted Reasoning —

Toshiro MINAMI, Hajime SANAMURA

(tos@ias.ias.fujitsu.junet)

International Institute for Advanced Study of Social Information Science (IIAS-SIS), FUJITSU LIMITED

と、要素を削除すること、基本要素を追加すること、要素を移動すること、要素を複写することなどの編集を行うことができる。

従来の証明・導出の編集機能は行編集に基づくことが多いが、落書き帳の上では推論規則の木構造をそのままの形態で編集する。上記の要素達は1つのオブジェクトとして表現され、それらを結合したり、また切り離したりの操作が自由にできる。

図1に落書き帳の上での証明の形態の例を示す。

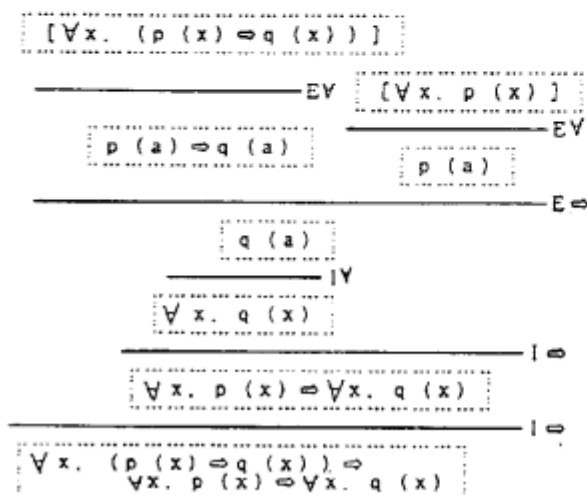


図1 落書き帳の上での証明の例

4. 証明の整理機能

落書き帳の上での表現様式は、試行錯誤的に証明を作り上げるのに適したものであり、結果として得られた証明は整理され見易い形ではない。これを、人間が通常用いる形式で表現し、定理、補題などの区別をして保存しておき、後の証明の際の検索、利用の便を図る。また、一旦整理した証明も、後でその証明を参考にするため等の用途で再び落書き帳の上で用いたいことがある。その場合は積書された証明を元の落書き帳での証明の形式に逆変換して用いることになり、そのための機能も必要である。

積書証明の例を図2に示す。

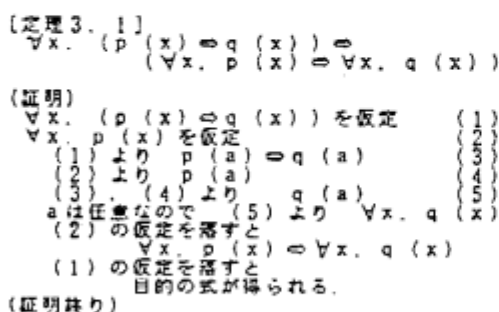


図2 証明の積書例

5. おわりに

本稿では計算機による論証支援システムの証明構築のための落書き帳の概念を中心に、論証支援システムが人間の論証をどのように認識し、それに対してどのような支援を行なおうとしているかを述べた。現在、推論機械PSI上への実現を想定した検討を行っている。論証支援システムの機能としては本稿で述べたような編集機能、情報検索機能などの一般的な支援機能の他、ある戦略に基づく証明の試行を行う機能、証明の簡略化の機能など、多くの論証支援特有の機能が考えられ、今後の課題である。

なお、論証支援システムについてのもっと詳しい説明およびこれまでの研究に関しては[6]～[9]を参照されたい。これらの文献における証明形式はいわば行編集の形式であり、本稿で述べた落書き帳のような画面編集の形式ではない。落書き帳においても論理式入力のためのツールとしての論理式エディタ[7]は依然有効であると考えられる。

謝辞

日頃御指導、御鞭撻をいただく北川会長、榎本所長に感謝いたします。なお、本研究の一部は第五世代コンピュータ・プロジェクトの一環としてICOTの委託で行ったものである。

参考文献

- [1] Ketonen, J. & Weening, J.S.: EXL - An Interactive Proof Checker, User's Reference Manual, Dept. of Computer Science, Stanford University, 1984.
- [2] Gordon, M.J., Milner, A.J. & Wadsworth, C.P.: Edinburgh LCF, LNCS, Springer-Verlag, 1979.
- [3] Aponte, M.V., Fernandes, J.A. & Roussel, P.: Editing First-Order Proofs: Programmed Rules vs. Derived Rules, Int. Symp. on Logic Programming, 1984.
- [4] Eriksson, A., Johansson, A.-L. & Tarnlund, S.-A.: Towards a Derivation Editor, Proc. of the 1st Int. Logic Programming Conf., 1982.
- [5] ICOT CAP-WG: CAPプロジェクト(1)～(6), 情報処理学会第32回全国大会, 1986年.
- [6] 沢村, 南, 佐藤, 小野, 小野: 論理式エディタ, 構造エディタに関するワークショップ, 1986年5月.
- [7] 佐藤, 小野, 小野, 沢村, 南: 論証支援システムのための論理式エディタ, 情報処理学会第33回全国大会, 1986年.
- [8] 南, 沢村: 論証支援のための証明コンストラクタ, 情報処理学会第33回全国大会, 1986年.
- [9] 南, 沢村: 論証支援システムの一構成, 日本ソフトウェア科学会第3回大会, 1986年.