

# 論証支援のための証明コンストラクタ

6L\_1

南 俊朗, 沢村 一

(富士通(株)・国際情報社会科学研究所)

## 1. はじめに

プログラムの作成、検証の他、様々な局面において何らかの対象モデルを構築し、そのモデルを操作することで問題解決を計ることが行われる。そのような対象モデルの表現の手段の1つとして、論理に基づくものが考えられる。我々は論理系に基づくモデルの構築・操作を計算機で支援するための論証支援システムについて研究を行っている。

問題領域に応じた論理系を定義し、その論理系の下での論証を支援するためには、その支援システムは論理系の定義機能、およびその論理系の下での証明に対する“編集”機能を持つようなメタ・システムが必要となる。本稿では、このようなメタ・システムへの第一歩として試作している証明の編集システム（証明コンストラクタ）について考察する。このコンストラクタは論理式の入力のためのツールとして、論理式入力編集機能を特に備えている。

第2節では本コンストラクタが証明の構成過程をどのようなイメージで捉えているのかについて説明する。第3節では証明コンストラクタの持つ種々の機能について概要を説明する。第4節では論理式入力支援機能について述べる。

## 2. 証明コンストラクト過程のモデル

図1に証明コンストラクタを利用して証明を構成する場面のイメージを示す。論理系は、何らかの方法で既に定められており、その論理系における証明がいくつか行われているものとする。それらの証明群を参照しながら新しい証明を構築している様子である。利用者はコマンドを用いて、推論規則や書き換え規則等を適用し、新たな式を生成していく。時には新しい式を入力したり、以前の式が不要になり削除したりの編集を繰り返す。このイメージはEKL[1]におけるものとはほぼ同じであるが、EKLの場合は書き換えによる式の変形を主な操作手段としている。

証明は原理的には自然演繹法と書き換え規則による書き換えを組み合わせた方法で行われ、論理的には導出の形態は木構

造となるが、本システムにおける具体的な表現上は式の線形な並びとして証明の状況が表現される。証明に属する各（論理）式は前提、公理、その証明内で以前に現れた式や、既に存在する別の証明内に現れている式に推論規則や書き換え規則を適用して得られる式のいずれかである。

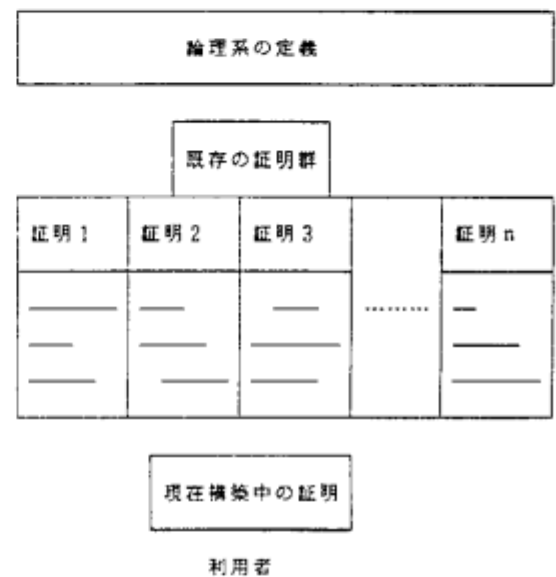


図1 証明コンストラクトの様子

## 3. 証明コンストラクト支援機能

本節では証明を構築するためのコマンドについて述べる。本コンストラクタのコマンドは大きく、記号の属性等の宣言、種々の情報表示、推論規則・書き換え規則の適用、式の削除・その取り消し等に分けることができる。記号の属性定義によって型、ソート、構文型、演算子等の指定が可能である。情報の表示機能としては記号の属性、証明中の式のリスト、式につけられたラベル等がある。推論規則はそれぞれに対応するコマンドによって行われる。たとえば、連言の導入(And Introduction)はコマンド

ai(<式の集まりの指定>, [<書き換え式>]),

Proof Constructor for the Reasoning Assistance

Toshiro Minami, Hajime Sawamura

International Institute for Advanced Study of Social Information Science,  
Fujitsu, Ltd.

によって行われる。このコマンド適用の結果、指定された式全部を連言記号(&)で結んだ式が作られ、もし、書き換え式が指定されているときは、さらに、それに従った書き換えが施され、その結果がコマンドの適用結果となる。書き換え式の指定は主に式に対する標準的な簡単化を行うために用いられる。

#### 4. 論理式入力支援機能

証明を構築するための機能としては証明編集機能のみで充分であるが、利用者が実際にシステムを用いる場合、比較的長い論理式を頻繁に入力する必要がある。このような利用者への負担をいくらかでも軽減する目的で論理式入力のためのエディタをフロント・エンド機能として用意した。論理式入力エディタのシステム全体に対する位置付けを図2に示す。

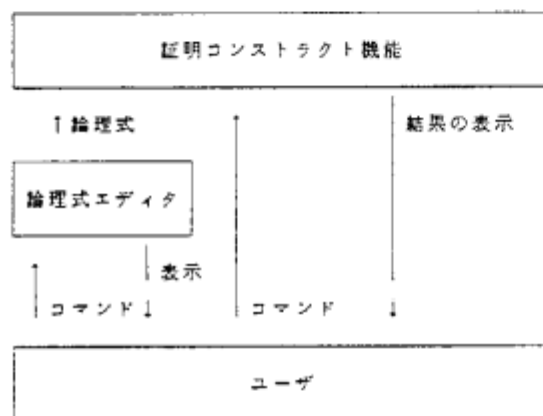


図2 論理式入力エディタの位置付け

本入力エディタは論理式の構造に対する木構造エディタとなっている。また簡単な式変形機能を持っており式の入力時に簡単な操作を行うことが可能である。

編集の対象となる論理式は図3に示すような形式で木構造が表示される。部分木の指定は「1-2-1」のようなノード番号によってなされる。ユーザの与えるコマンドによって

```

1 all(x, x+1=add1(x))
|----- 1-1 x
|----- 1-2 x+1=add1(x)
|         |----- 1-2-1 x+1
|         |         |----- 1-2-1-1 x
|         |         |----- 1-2-1-2 1
|         |         |----- 1-2-2 add1(x)
|         |         |----- 1-2-2-1 x

```

図3 式  $\text{all}(x, x+1=\text{add1}(x))$  の木構造表示

部分木に対する挿入、削除、移動、複写、変換、置換等の木構造編集操作が可能である。また論理式に対する固有なコマンドとして変数への式の代入、論理的同値式への簡約化機能を備えている。

#### 5. おわりに

本稿では計算機による論証支援システムの目途点、そのための証明コンストラクト支援機能、および論理式の入力支援のためのエディタについてその概略を述べた。現在VAX/11-780上のC-PROLOGの下で試作したシステムをPSI上のESP版に移植中である。ESP版に対する課題として、ビットマップ・ディスプレイによるマルチ・ウィンドウの環境を活かしたユーザ・インタフェースの実現、たとえば、木構造の表示、部分木の指定方法の改善等がある。もちろん、論証支援システムとしては、論理系の定義機能の拡張等、基本的課題は多い。今後、このような点について機能の改善および追加を行っていく予定である。

なお、論理式エディタに関しては[5]、[6]も参照されたい。また、既存の証明・推論エディタとして[2]～[4]等が発表されている。

#### 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.: EKL - An Interactive Proof Checker, User's Reference Manual, Dept. of Computer Science, Stanford University, 1984.
- [2] Aponte, M.V., Fernandes, J.A., Roussel, P.: Editing First-Order Proofs: Programmed Rules vs. Derived Rules, Int. Symp. on Logic Programming, 1984.
- [3] Eriksson, A., Johansson, A.-L., Tarnlund, S.-A.: Towards a Derivation Editor, Proc. of the 1st Int. Logic Programming Conf., 1982.
- [4] 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.
- [5] 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.
- [6] 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, 本予稿集.