

2C-6

論証支援システムのための証明支援 -思考シートによる証明方法-

横田かおる 大橋恭子 沢村一 南俊明
(富士通株式会社)

1. はじめに

今日、計算機科学や人工知能の分野において、計算機上での定理証明およびその支援の研究が盛んである。論理の下での問題解決は、形式的な手続きによって行うことができるため、有効であることが認められている。

問題解決を論理の下で行うには、解決する問題に適した論理系の下で行うことが望ましい。しかしながら、問題毎に適切な論理系の下での支援システムを個々に構築するのは、その度毎に類似の労力を要し、無駄が多い。そのため、論理系を定義するためのメタな枠組みを提供し、ユーザの定義した論理系の下での支援を行うようなシステムが有効である。我々は、このような要求を満たす汎用論証支援システムEUODHILOS¹⁾(Every Universe Of Discourse Has Its Logical Structure²⁾)を開発し、その一部として証明支援部を提供した。

我々は、ユーザが定義した論理系の下での論証を進めるための道具となるような証明支援部を開発することを目標とした。本証明支援部は、人間の思考に沿った支援、証明に適したユーザ・インタフェースを特徴とする。本システムでは、思考の道具としての役割を重視し、証明支援部を思考シートと呼ぶ。

2. 論理

思考シートでは、証明を自然演繹法³⁾スタイルで行う。証明を行う論理系の定義は、ユーザがEUODHILOSの論理系定義部で次の定義を記述することで定められる。

- (1) 記号定義 : それぞれの論理に特有な記号(∧, ∨ など)をフォント・エディタで作成する。
- (2) 構文定義 : 論理式や項の構文を確定節文法で定義する。この定義をもとに論理式のためのパーザが生成される。
- (3) 公理定義 : 証明の出発点となる公理や公理図式を定義する。公理図式は、証明中、図式のまま、あるいは具体化して使用することができる。思考シートでは、公理以外に仮定を証明の出発点とすることもできる。
- (4) 導出規則 : 導出規則としては、推論規則と書き換え規則をそれぞれ、以下のように定義する。

・推論規則定義

[仮定₁] … [仮定_n]

前提₁ … 前提_n
結論

・書き換え規則定義

書き換え前表現
書き換え後表現

推論規則には、必要に応じて適用条件を付けて定義する。規則は論理図式で記述され、証明中、具体化されて適用される。適用は、規則各々に付ける名前で指定する。図1に論理系定義と証明の関係を示す。論理系定義は、証明の際に参照される。論理系定義とこの論理系の下で行った証明、導かれた定理などは合わせて理論として理論データベース内で管理される。証明は、1つの理論の下で、複数個作成することができ、完成されていない証明断片や途中の状態でも保存することができる。

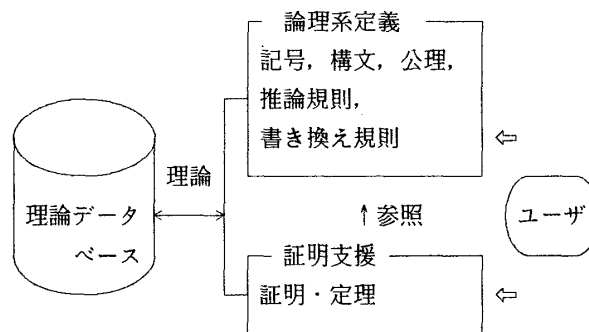


図1 論理系定義と証明

3. 人間の思考に沿った支援

証明の過程は、問題を適切に解決するための証明を得るまでの試行錯誤の過程である。我々は、このような過程を支援することを重視し、証明全体のチェックを一括して行うような証明チェッカーではなく、証明の1ステップずつを支援する対話型の証明コンストラクタを開発した。

我々は、思考シートが、ユーザの思考に沿って、実験、修正を容易に行える環境を提供することを目標とした。すなわち、ユーザが好きなのところから証明に手をつけ、好きな方法で考え、やり直しが容易にできる環境構築を目指した。この条件を満たすため、思考シートでは、次のような証明方法が可能である。

An proof constructor for a general-purpose reasoning assistant system
- Proof methods in Sheets of Thought -

Kaoru YOKOTA, Kyoko OHASHI, Hajime SAWAMURA, Toshiro MINAMI
Fujitsu, Ltd.

(1) 前向き導出, 後ろ向き導出

前提, 結論, 導出規則名の3つのうちの2つが与えられると, 定義された規則を参照して他の1つを導く. すなわち, 前提と規則名から結論を求める前向き導出, 結論と規則名から前提を求める後ろ向き導出の両方と, 前提と結論から規則名を求めることが可能である.

(2) 証明の結合・分離

証明を中間結果まで進めたり, 逆に途中から始めたりして, 部分証明を作成し, 後でそれらを結合して全体証明を完成する方法や, 逆に, より大きな証明を分割して部分証明を利用する方法など, 人間が用いている様々な思考方法を実現できる.

(3) 不完全部分の保留

思いついた時点ですべての情報が完全であるとは限らないので, 思考シートでは, 不完全な部分を保留して, “とりあえず”証明を進め, 後で不整合部分の条件を満足させて, 証明を完成させるといった証明方法が可能である. また, 詳細に分からない論理式をメタ変数を使用して記述し, 変数部分を後で具体化することも可能である.

(4) 容易な修正

導出の取消や不適切な論理式の削除によって, 証明のやり直しが容易に行える. これによって, 証明が正しくなくなる場合は, 整合性を保つように証明が分離されたり, 不整合であることが表示される.

4. 証明に適したユーザ・インタフェース

思考に沿った支援を実現するには, それと同時にシステムの使いにくさが原因となる思考の中断を防止し, 更に思考を促すようなインタフェースを提供することが望ましい. そのため, 思考シートは, 次のようなインタフェースを持つことを特徴とする.

(1) 複雑な証明, 論理式の図表示

複雑な証明や論理式の構造がユーザに分かり易く図で表示される. 証明は, 公理や仮定をリーフ, 結論をルートとする木構造を持っているので, 図2のように証明図³⁾で表示する. 論理式は, その木構造を図3のように表示する.

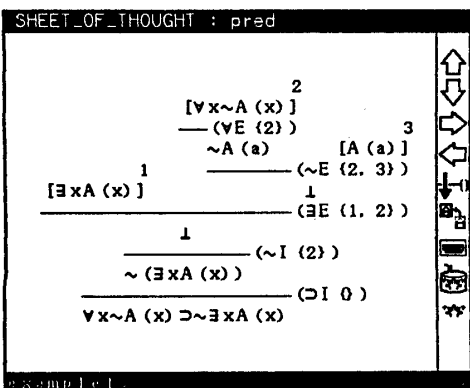


図2 証明図

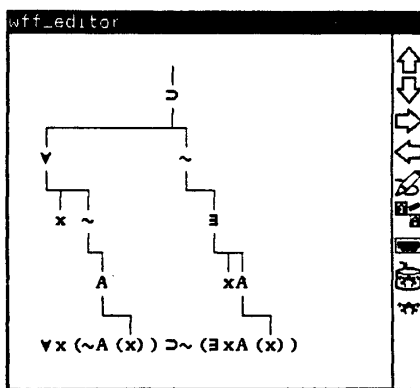


図3 論理式の木構造表示

(2) 証明(断片)のグループ化

マルチ・ウィンドウ環境を利用して, 関連のありそうな証明(断片)を同じウィンドウに表示するなど, 複数ウィンドウをメモ用紙のように使うことができ, 更にウィンドウ間の証明断片, 論理式の移動や複写が容易に行える.

(3) マウス主体のコマンド

操作する論理式や証明断片の指定が, 図の上でマウスを用いて行うことができる. コマンドは, マウスで指定された論理式や証明断片と, 表示されているアイコンや選択されたメニューの項目によって決定される. そのため, 煩わしいコマンド行の入力などがなく, 証明の操作が容易に行える.

(4) 不完全部分の表示

条件が整わないために, 不完全なまま証明が進められている導出ステップに印*を表示して, 正しい導出の行われたステップと区別する.

5. おわりに

本稿では, 論証支援システムEUODHILOSにおける証明支援部(思考シート)について報告した. システムの有効性を確認するために, 現在までに, 述語論理, 様相論理, 内包論理, Hoare 論理, dynamic 論理などの定義を行い, その下での証明を行った. 1つの論理の定義および証明を数時間程度で行うことができる.

思考シートは, 使いやすさに重点を置いて設計を行った. しかし, ユーザの操作に制約が少ないため, 証明全体の整合性をチェックするタイミングなどが難しく, 今後の課題となっている.

今後, 更に様々な論理に対して実験を行い, 使用経験を通して更に改良を行う予定である.

なお, 本研究は, 新世代コンピュータの開発の一環として, ICOTの委託によって行ったものである.

参考文献

[1] 南他: 論証支援システム: 論理モデル構築のための支援ツール, Proceedings of the Logic Programming Conference 88', ICOT(1988)
 [2] Langer, S.K.: A set of postulates for the logical structure of music, Monist 39, pp.561-570(1925)
 [3] 前原昭二: 記号論理学入門, 日本評論社(1967)