数理解析研究所講究録 1375 巻 2004 年 253-259

人間指向型汎用類推証明システムの開発

尹 淑萍† 山田 敬三‡ 平田 耕一‡ 原尾 政輝‡
† 九州工業大学大学院情報工学研究科
‡ 九州工業大学情報工学部

Yin Shuping† Keizo Yamada‡ Kouichi Hirata‡ Masateru Harao‡
† Graduate School of Computer Science and Systems Engneering
‡ Department of Artificial Intelligence Kyushu Institute of Technology

1 まえがき

自動証明システムは**機械指向型** (*machine-oriented*) と**人間指向型** (*human-oriented*) に分けられる. 機械指向型は導出原理, タブロー法, コネクション法, Boyer Moore 証明器などに代表される. それらの特徴は効率を追求することを主目的にしており, 証明の過程はあまり問題にしない. 一方, 人間指向型は人間の思考過程を重視したシステムで, **証明検査器** (*proof checker*), **証明支援** (*proof planner*) などに代表されるように, その証明過程を問題にし, 人間にとって分かり易い証明を実現することが求められる. 人間指向型は人間の行う証明に近づくので, 教育支援を始め広い範囲の応用が期待できる.

人間指向型証明器を実現するには, **シークェント計算** (*sequent calculus*) や**自然演繹** (*natural deduction*) といった証明システムを直接機械化すればよい. しかしながら, その証明の過程は発見的な要素が強く, 一般に自動化は困難である. そのため, 本研究ではすでに証明の分かっている論理式との類似性から証明制御情報を抽出しそれを援用して証明する, **スキーマ誘導類推**(*schema-guided analogy*) を提案して自動化する. ここで類推とは, 例えば我々がある問題を解こうとしたときに, 既に解いたことのある, それと似た問題を参考にしながら解くような推論方式である.

本論文では, 述語論理式の証明を, 自動生成することを目標とするので, 証明構造の類似性に基づいた論理式間の類似性を定義する. このような証明構造の類似性を記述するのに**スキーマ** (*schema*) の概念を用いる [5, 6, 12, 13, 14]. スキーマとは一般化された知識で, 本論文では証明可能な 2 階述語論理式として定義する.

本研究で提案するスキーマ誘導類推証明は, 証明の再利用とスキーマ誘導処理を融合して類推を実現するものである. 特に, スキーマとして証明可能な 2 階の論理式を用いることによってスキーママッチングによる正当性の保証された証明の導出を実現することができる.

さまざまな論理体系の証明を支援する対話型定理証明システムには, Isabelle[8, 9, 10] や Isabelle に Tcl/TK による GUI(*Graphical User Interface*) を付加した XIsabelle[2] などがある. しかし, これらのシステムでは, 高度な汎用性を持っている反面, ユーザインターフェースの機能が犠牲となっており, 操作が複雑で扱いづらいという欠点を持っている. 一方, 教育的観点からは, 初学者がシステムを扱うことを考慮して, 証明に対して理解を深めやすいように, 証明図の表示, 証明の実行, 入力に関してのユーザインターフェースが充実した定理証明システムを構築する必要がある. 本研究は, これらの機能を持つ汎用的な証明システムを開発することも目的としている.

本稿の構成は次の通りである. 2 節では, 論理式と 2 階マッチングについて, 3 節では, 証明システムについて説明する. 4 節では, 類似性, スキーマ誘導類推について述べる. 5 節では, システム構成と実行例を示す. 6 節は考察である.

2 基本的定義

この節では, 以後の議論で用いる基本的な定義を与える. 詳細は文献 [6, 13] を参考すること.

2.1 論理式

論理式は通常の 1 階述語論理式に述語変数を含む 2 階の論理式を対象とする. そのため, それぞれ可算の**個体定数** (*individual constant*) の集合を IC (要素を $a, b, c, \ldots$ で表す), **個体変数** (*individual variable*) の集合を IV (要素を $x, y, z, \ldots$ で表す), **関数定数** (*function constant*) の集合を FC (要素を $f, g, h, \ldots$ で表す), **述語定数** (*predicate constant*) の集合を PC (要素を $p, q, r, \ldots$ で表す) 及び**述語変数** (*predicate variable*) の集合を PV (要素を $P, Q, R, \ldots$ で表す) を仮定する.

特に, ここでの述語変数は 2 階の変数 [4, 5] とする.

FC, PC, PV の要素 d の引数の数が n のとき，d は n 引数を持つという．

定義 1 項 (*term*) を次のように帰納的に定義する．

1. 個体定数 $c \in IC$ と個体変数 $v \in IV$ は項である．
2. f が n 引数の関数定数 $(n \geq 0)$ かつ，$t_1, \cdots, t_n$ が項ならば，$f(t_1, \cdots, t_n)$ は項である．

定義 2 論理式 (*formula*) を次のように帰納的に定義する．

(1) $\alpha \in PC \cup PV$ が引数 n を持ち，$t_1, \cdots, t_n$ が項ならば，$\alpha(t_1, \cdots, t_n)$ は論理式である．$\alpha(t_1, \cdots, t_n)$ を**原子論理式** (*atomic formula*) という．特に，$\alpha \in PV$ のとき 2 階の原子論理式という．
(2) A, B が共に論理式ならば，$A \wedge B, A \vee B, A \supset B, \neg A$ はいずれも論理式である．
(3) A が論理式で，x が個体変数ならば，$\forall x.A, \exists x.A$ はともに論理式である．

論理式に対する自由変数と束縛変数の定義は，一般の 1 階論理と同様に定義する．ここで定義した 2 階論理式は，2 階の述語変数は含むが，関数変数と固定自由変数は含まない事に注意する．また，述語変数は自由な出現だけを仮定する．例えば，

$$\forall x.P(x,a,b) \vee (\forall y.P(y,b,y))$$

は 2 階論理式の例で，P が述語変数である．

2.2 代入

2 階論理式を $\Phi, \Psi, \ldots$，1 階閉論理式を $\phi, \psi, \ldots$ と表す．代入 θ とは，IV から項集合への写像，PV から論理式集合への写像よりなり，$\theta = [x := t, \ldots, P := p, \ldots]$ で表す．

定義 3 式 t に対し，$t\theta$ を次のように定義する:

(1) $t = c, c \in IC$ のとき，$t\theta = c$.
(2) $t = x, x \in IV$, のとき，$[x := t'] \in \theta$ ならば $t\theta = t'$，そうでなければ $t\theta = x$ になる．
(3) $t = f(t_1, \ldots, t_n)$ のとき，$t\theta = f(t_1\theta, \ldots, t_n\theta)$.
(4) $t = P(t_1, \ldots, t_n)$, $P \in PV$, のとき，$[P := p(f_1(y_1, \ldots, y_n), \ldots, f_k(y_1, \ldots, y_n))] \in \theta$ ならば，$t\theta = p(f_1(y_1, \ldots, y_n)\sigma, \ldots, f_k(y_1, \ldots, y_n)\sigma)$．ただし，$\sigma = [y_1 := t_1\theta, \ldots, y_n := t_n\theta]$ なる代入．
(5) $t = Qx.p$ $(Q \in \{\forall, \exists\})$ のとき，$t\theta = Qy.(p\theta)$.

定義 3 の (4)，(5) では，代入の際，変数条件などを損なわないようにする [13]．例えば，(5) で代入の結果 Qx で束縛される場合は新しい変数 y を用いて名前の付け替えを行い $t\theta = Qy.p((p\{x := y\})\theta)$ などとする．2 階の論理式 Φ にある代入 θ を適用して得られる論理式を Φ の**インスタンス**(*instance*) と呼ぶ．

例 1 2 階論理式 Φ の代入 θ の下でのインスタンス:

$$\Phi = (\forall x.P(x,a,b)) \vee (\forall y.P(y,b,y)),$$
$$\theta = [P := \exists y.p(v_3, y, v_2)].$$

このとき，インスタンス $\Phi\theta$ は以下のようになる:

$$\begin{aligned}
\Phi\theta &= ((\forall x.P(x,a,b) \vee (\forall y.P(y,b,y))\theta \\
&= (\forall x.(P(x,a,b))\theta \vee (\forall z.(P(z,b,z))\theta \\
&= \forall x.(\exists y.p(v_3,y,v_2)[v_1 := x, v_2 := a, v_3 := b]) \\
&\quad \vee \forall z.(\exists y.p(v_3,y,v_2)[v_1 := z, v_2 := b, v_3 := z]) \\
&= (\forall x.\exists y.p(b,y,a)) \vee (\forall z.\exists y.p(z,y,b)).
\end{aligned}$$

2.3 2階マッチング

2 階論理式 Φ と閉じた 1 階論理式 ϕ の対 $\langle \Phi, \phi \rangle$ から，$\Phi\theta = \phi$ となる代入 θ を求める操作を 2 階マッチングという．また，θ が存在するか否かを決定する問題を**2 階マッチング問題** (*second-order matching problem*) という．2 階マッチング問題は一般に NP-完全であることが知られている．しかし，Φ は関数変数を含まず述語変数も先頭にのみ出現する 2 階論理式であり，ϕ は閉じた 1 階論理式という条件があるため，効率的な 2 階マッチングアルゴリズムが構成可能である．

命題 1 ([13]) Φ を 2 階論理式，ϕ を閉じた 1 階論理式とする．このとき，2 階マッチング問題は多項式時間で解ける．

特に，後の証明制御情報を取り出す目的のためには，すべてのマッチング代入を求める必要はなく，証明生成のために適したものを生成できればよい．そのため，最適なマッチングを効率的に求める手法を用いる [6].

3 証明システム

3.1 シークェント計算

Γ と Δ を論理式の集合 $\{A_1, \cdots, A_m\}, \{B_1, \cdots, B_n\}$ とする．このとき，Γ, Δ の要素の列の組

$$A_1, \cdots, A_m \vdash B_1, \cdots, B_n$$

を**シークェント** (*sequent*) という．これを簡単に $\Gamma \vdash \Delta$ とも表す．シークェント $\Gamma \vdash \Delta$ の真理値は，$A_1 \wedge \cdots \wedge A_m \supset B_1 \vee \cdots \vee B_n$ で定義する．特に，$\vdash$ の左右に同じ論理式が現れるシークェントを**公理** (*axiom*) と

いう. S_1, S_2, S_3 をそれぞれシークェントとする. このとき, **推論規則** (*inference rule*) とは, 次の形をした図である:

$$\frac{S_1}{S}, \quad \frac{S_1 \quad S_2}{S} \text{ または } \frac{S_1 \quad S_2 \quad S_3}{S}.$$

推論規則は, 直観的には $S_1(S_2,S_3)$ から S が導出されることを表している. また, 論理式 ϕ だけから成る $\vdash \phi$ を ϕ のシークェントという.

シークェント計算は推論規則と公理からなる. 論理式 ϕ の証明は, そのシークェントから出発して順次推論規則を適用して公理へ分解する過程よりなり, それはシークェントを節とする木構造の**証明図**(*proof figure*) で表すことができる. このとき ϕ をゴールという. ある論理式 ϕ に対して, ϕ の証明図の葉がすべて公理となるとき, ϕ は**証明可能** (*provable*) という.

シークェント計算は, 述語論理のための LK と LJ が一般的であるが, シークェントに適当な制限をつけたり推論規則を定義することによって様々な論理体系を表すことができる. また, シークェント計算はゴール (証明したい論理式) を分解する形で証明が進むので対話的証明に向いており, 自動化もしやすい. そのため, 自然演繹体系 NK と NJ もシークェント形式を用いた証明法を用い, 通常の自然演繹証明はシークェント形証明から変換して求める形式をとった. また, 本研究で提案する手法が様相論理や線形論理といった非標準論理でも適用可能であることを示すため, 代表的な様相命題論理である体系 S4 について実装した. 次にそれらの体系の特徴や実装に工夫などについて述べる.

3.2 LK と LJ

シークェント計算 LK の推論規則にはいくつかの体系がある [11] が, 本論文では, [14] の 12 規則を用いる. そこで $\vee$ right としては次の規則を用いている.

$$\text{LK: } \frac{\Gamma \vdash \Delta, A, B}{\Gamma \vdash \Delta, A \vee B}(\vee\text{right}).$$

この規則は後述する LJ と同じ規則を用いてもできるが, ここでは LK と LJ との違いを明示的に示すため, また構造規則などの煩わしさを省き証明を簡略化するためにこの規則を採用した. 直観主義論理の体系 LJ では, シークェントの $\vdash$ の右側には高々1 つの論理式しか許さない制約があるので, LJ の推論規則は規則 $\vee$ right の代わりに次の二つの規則を用いる.

$$LJ: \frac{\Gamma \vdash A}{\Gamma \vdash A \vee B}(\vee\text{right 1}), \quad \frac{\Gamma \vdash B}{\Gamma \vdash A \vee B}(\vee\text{right 2}).$$

実装では $\vee$ right の推論規則の適用の選択ができるような機構を工夫する.

構造規則も導入して厳密な扱いをすることも可能であるが, 煩雑さを避けるためここでは次のカット規則のみを実装した.

$$\frac{\Gamma \vdash \Delta, A \quad A, \Pi \vdash \Sigma}{\Gamma, \Pi \vdash \Delta, \Sigma}(\text{cut}).$$

カット規則によって, より柔軟な証明が実現可能となり, 教育支援などへの応用ではカット規則を用いた場合と用いない場合の証明の違い等を明示的に示すことができる. カット規則は「証明に用いる上辺の部分論理式は必ず下辺にも出現する」という**部分論理式条件**(*subformula property*) を満たさないため, 自動化は一般に困難である. そのため, 対話証明の場合は手動でユーザが必要な情報を指定できるようにする. スキーマ誘導類推では他の規則と同様に自動証明が可能である.

例 2 論理式 ϕ の LK 証明

$\phi = (\forall x.(A \supset p(x)) \supset (A \supset \forall x.p(x)))$,

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{A \vdash p(z), A \quad p(z), A \vdash p(z)}{A \supset p(z), A \vdash p(z)}(\supset \text{right})}{\forall x.(A \supset p(x)), A \vdash p(z)}(\forall\text{left})}{\forall x.(A \supset p(x)), A \vdash \forall x.p(x)}(\forall\text{right})}{\forall x.(A \supset p(x)) \vdash A \supset \forall x.p(x)}(\supset \text{right})}{\vdash (\forall x.(A \supset p(x)) \supset (A \supset \forall x.p(x)))}(\supset \text{right})$$

例 3 例 2 の論理式 ϕ の LJ 証明

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{A \vdash A \quad p(z), A \vdash p(z)}{A \supset p(z), A \vdash p(z)}(\supset \text{right})}{\forall x.(A \supset p(x)), A \vdash p(z)}(\forall\text{left})}{\forall x.(A \supset p(x)), A \vdash \forall x.p(x)}(\forall\text{right})}{\forall x.(A \supset p(x)) \vdash A \supset \forall x.p(x)}(\supset \text{right})}{\vdash (\forall x.(A \supset p(x)) \supset (A \supset \forall x.p(x)))}(\supset \text{right})$$

3.3 シークェント形式自然演繹証明

自然演繹の体系 NK や NJ[1] は, トップダウンに仮定から結論に至るまでの推論を行う. この方式は各ステップで用いる仮定や規則などを機械的に与えることが難しい. そのため, 本研究ではこれらの体系のシークェント形式での証明方式である NK*, NJ*を採用した. シークェント形式での証明は, 下辺から上辺へ進むので用いる推論規則や仮定などがより機械的に求まる. そして, NK*, NJ*上の証明から NK, NJ 証明を簡単な変換で導くことができる. NK*, NJ*の証明には [11] の推論規則を用いる.

例 4 例2の論理式 ϕ のNK*証明を求める．証明は下式に推論規則を適用し上式を導出する．そして，シークェントの左辺には $\supset$ I で用いられた仮定が記述されていることに注意する．この図では，全ての葉が公理になるので ϕ は証明可能である．

$$\dfrac{\dfrac{\dfrac{A\,①\vdash A \quad \dfrac{\forall x.(A\supset p(x))\ ②\vdash \forall x.(A\supset p(x))}{\forall x.(A\supset p(x))\ ②\vdash A\supset p(z)}(\forall \mathrm{E})}{\forall x.(A\supset p(x))\ ②, A\ ①\vdash p(z)}(\supset \mathrm{E})}{\forall x.(A\supset p(x))\ ②, A\ ①\vdash \forall x.p(x)}(\forall \mathrm{I})}{\forall x.(A\supset p(x))\ ②\vdash A\supset \forall x.p(x)}(\supset \mathrm{I})}{\vdash \forall x.(A\supset p(x))\supset (A\supset \forall x.p(x))}(\supset \mathrm{I})$$

一方NKにおける ϕ の証明は次のようになる．ここで A① 等のラベルは，$\supset I$ 規則によって仮定式が消去された事を示す．この場合，ϕ を終式とする変換が存在し，すべての仮定が除去されたので，ϕ は証明可能である．

$$\dfrac{\dfrac{\dfrac{A\,① \quad \dfrac{\forall x.(A\supset p(x))\ ②}{A\supset p(z)}(\forall \mathrm{E})}{p(z)}(\supset \mathrm{E})}{\forall x.p(x)}(\forall \mathrm{I})}{A\supset \forall x.p(x)}(\supset \mathrm{I}①)}{\forall x.(A\supset p(x))\supset (A\supset \forall x.p(x))}(\supset \mathrm{I}②)$$

NK*とNKの証明を比べると，NK*の右辺だけをみればNK証明になっていることが分かる．ただし，左辺の仮定がすべて右側に出現していない場合があり，その場合は仮定を導入する適当な場所（最初でも構わない）を指定する必要がある．この単純な変換でNK*証明からNK証明が得られる．

例 5 NK*からNKへの変換
$(A\supset (B\supset A))$,$(A\supset ((A\wedge B)\supset B))$ のNK*証明（左）とNK証明（右）は次のようになる．

$$\dfrac{\dfrac{B\,①, A\,②\vdash A}{A\,②\vdash B\supset A}(\supset \mathrm{I})}{\vdash A\supset (B\supset A)}(\supset \mathrm{I}),\qquad \dfrac{\dfrac{B\,①, A\,②}{B\supset A}(\supset \mathrm{I}①)}{A\supset (B\supset A)}(\supset \mathrm{I}②),$$

$$\dfrac{\dfrac{\dfrac{A\,②, A\wedge B\,①\vdash A\wedge B}{A\,②, A\wedge B\,①\vdash B}(\wedge \mathrm{E1})}{A\,②\vdash (A\wedge B)\supset B}(\supset \mathrm{I})}{\vdash A\supset ((A\wedge B)\supset B)}(\supset \mathrm{I}),\qquad \dfrac{\dfrac{\dfrac{A\wedge B\,①}{B}(\wedge \mathrm{E1})}{(A\wedge B)\supset B}(\supset \mathrm{I}①)}{A\,②\supset ((A\wedge B)\supset B)}(\supset \mathrm{I}②)$$

3.4 非標準論理

一般に古典論理以外の論理を非標準論理と呼ぶが，代表的なものに様相論理や線形論理などがある．これらの論理では，新たに論理演算記号を導入する必要がある．本論文では様相命題論理S4について述べるが，他の論理に関しても同様な手法で自動化が可能である．ただし，2階マッチングにおいては論理記号は定数として扱い，意味的な等価変換などは考慮しない．

様相命題論理は命題論理に様相記号を加えたものである．ここでは様相記号として**必然演算子** $\Box$(*necessity*) のみを扱い，**可能演算子** $\Diamond$(*possibility*) は構文のレベルで $\neg\Box\neg$ の略として扱う．

S4の推論規則は，LKの推論規則の命題の8個 [14] に次の2つを加えたものである：

$$\dfrac{A,\ \Gamma\vdash \Delta}{\Box A,\ \Gamma\vdash \Delta}(\Box \mathrm{left}),\qquad \dfrac{\Box\Gamma\vdash A}{\Box\Gamma\vdash \Box A}(\Box \mathrm{right}).$$

例 6 S4における次の命題 $((\Box A\wedge \Box B)\supset \Box(A\wedge B))$ の証明は次のようになる．

$$\dfrac{\dfrac{\dfrac{A,\Box B\vdash A}{\Box A,\Box B\vdash A}(\Box \mathrm{left}) \quad \dfrac{\Box A, B\vdash B}{\Box A,\Box B\vdash B}(\Box \mathrm{left})}{\Box A,\Box B\vdash A\wedge B}(\wedge \mathrm{right})}{\Box A,\Box B\vdash \Box(A\wedge B)}(\Box \mathrm{right})}{\Box A\wedge \Box B\vdash \Box(A\wedge B)}(\wedge \mathrm{left})}{\vdash (\Box A\wedge \Box B)\supset \Box(A\wedge B)}(\supset \mathrm{right})$$

4 スキーマ誘導類推

4.1 スキーマ

スキーマ誘導類推 [5] は，「似た論理式は似た証明を持つ」という証明方式を定式化したものである．スキーマはそれを用いて構成された証明はまた正しいという正当性を保証するものでなければならない．そのため証明可能性が保証された2階の論理式として定義する．

定義 4 スキーマとは証明可能な2階論理式である．

スキーマとして何を蓄積するかでシステムの自動証明能力も変ってくる．そのため，スキーマは対話証明などで証明された論理式を一般化して蓄積していく機能を持たせる．いま論理式を ϕ，その証明を $proof(\phi)$ で表す．そのとき次の性質が成り立つ．

命題 2 ϕ を証明 $proof(\phi)$ を持つ論理式とする．ϕ に含まれる述語定数を2階述語変数に置き換えて得られた2階論理式を Φ とすると，Φ は証明 $proof(\Phi)$ を持つスキーマである．

(証明) $proof(\Phi)$ は $proof(\phi)$ における述語定数を述語変数に置き換えただけであるから証明図の葉は公理になっている．従って，述語変数への代入が変数条件などを満たす範囲で $proof(\Phi)$ もまた証明になるが，ここで仮定する代入は2階マッチングであり変数条件などは損なわない．すなわち，Φ は証明 $proof(\Phi)$ を持つスキーマとすることができる． □

証明可能なϕから述語を変数化して構成されたスキーマΦを，ϕから構成されたスキーマという．

例 7 論理式を$\phi = ((p(a) \vee q(b)) \wedge \forall x.(p(x) \supset q(x))) \supset \exists x.q(x)$とする．その証明$proof(\phi)$は次のように(図1)与えられる．

ϕの述語定数を2階の述語に変換して得られた2階論理式をΦとする：

$$\Phi = ((P(a) \vee Q(b)) \wedge \forall x.(P(x) \supset Q(x))) \supset \exists x.Q(x)$$

ただし，ここでP, Qは2階の述語変数である．このとき，$proof(\phi)$から述語定数を述語変数に変換して得られる次の証明図(図2)はΦの証明$proof(\Phi)$となる．

Φをスキーマとする．いま，$\psi = \Phi\theta$なる任意のΦのインスタンスの証明$proof(\psi)$を考える．$proof(\Phi)$にマッチングで得られた代入θを行った証明を$(proof(\Phi))\theta$で表すとき，次の関係がある．

命題 3 Φをスキーマ，ψを$\psi = \Phi\theta$なる任意のΦのインスタンスとする．このとき，$proof(\psi) = (proof(\Phi))\theta$の関係が成り立つ(図3)．

(証明) シークェント$\Gamma \vdash A$が証明可能であるならば，トートロジー定理によって述語変数に論理式を代入したシークェント$\Gamma[P := p(x_1, \cdots, x_n)] \vdash A[P := p(x_1, \cdots, x_n)]$もまた証明可能である．ただし，ここで$\Gamma[P := p(x_1, \cdots, x_n)] \vdash A[P := p(x_1, \cdots, x_n)]$は$\Gamma, A$中の$P(t_1, \cdots, t_n)$を$p(x_1, \cdots, x_n)$で置き換えたシークェントで，この代入は2節の定義3に基づいて行うので，論理的な正しさを保証している． □

4.2 類似性とスキーマ誘導類推証明

命題3より，あるスキーマのインスタンスは同じ証明図を持っている．従って，スキーマを用いて論理式間の類似性を次のように定義する．

定義 5 類似性: 論理式ψとϕが共にあるスキーマΦのインスタンスとなっているとき，ψとϕはスキーマΦの下で類似しているという(図3)．

スキーマと1階の閉論理式とのマッチングを**スキーママッチング**(*schema matching*)とよび，論理式間の類似性は同じスキーマとマッチング可能かどうかで決定できる．

スキーママッチングは論理的意味を保存する代入を導出する[4]ので次の性質が成り立つ．

定理 1 スキーマΦの下で類似するインスタンスを$\phi(= \Phi\theta)$，$\psi(= \Phi\rho)$とする(図3)．このとき，ϕ，ψは，類似の証明$(proof(\Phi))\theta$，$(proof(\Phi))\rho$を持つ(図3)． □

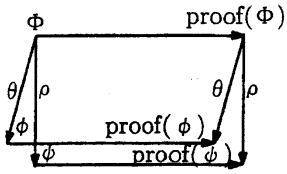

図 3: スキーマ誘導類推

スキーマ誘導類推では，与えられた問題を直接解くのではなく，以下の方法に沿って証明を導出する．まず，既知の問題とその解を抽象化したスキーマと証明スキーマの組を**スキーマベース** (***schema base***)$[(\Phi_i, proof(\Phi_i))|i \in I]$に蓄えておく．未知の問題$\phi$が与えられたとき以下の手順と適用する．

---

function *AnalogyBasedProving*;
input ϕ;
  search Φ and match $\Phi\theta = \phi$;
if ϕとスキーマΦがマッチング可能 **then do**
  $proof(\Phi)\theta$によりϕの証明を導出する;
  return $proof(\Phi)\theta$;
else do begin
  手動で証明を提示する;
  結果$(\Phi_i, proof(\Phi_i))$をスキーマベースに格納する;
end of function

図 4: スキーマ誘導類推証明の手続

例 8 $\psi = ((r(a) \vee s(b)) \wedge \forall x.(r(x) \supset s(x))) \supset \exists x.s(x)$

の証明を求める．まず，マッチング可能なスキーマを探索する．ψが例8のスキーマΦとマッチング可能なことを発見し，スキーマΦとψのマッチング代入θを求める．ここで

$$\theta = [P := r(y), Q := s(z), y := a, z := b]$$

θを$proof(\Phi)$(図2)に適用して$proof(\psi)$は$proof(\Phi)\theta$として求まる．例えば，図2における$(*)$印の論理式についての証明図は次のようになる．

$$\dfrac{\dfrac{r(a) \vdash r(a) \quad \dfrac{s(a) \vdash s(a)}{s(a) \vdash \exists x.s(x)}\,(\exists \text{right}[x := a])}{r(a), r(a) \supset s(a) \vdash \exists x.s(x)}\,(\supset \text{left})}{r(a), \forall x.(r(x) \supset s(x)) \vdash \exists x.s(x)}\,(\forall \text{left}[x := a]).$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{p(a)\vdash p(a)\quad \dfrac{q(a)\vdash q(a)}{q(a)\vdash \exists x.q(x)}\,(\exists\mathrm{right}[x:=a])}{p(a),p(a)\supset q(a)\vdash \exists x.q(x)}\,(\supset\mathrm{left})}{p(a),\forall x.(p(x)\supset q(x))\vdash \exists x.q(x)}\,(\forall\mathrm{left}[x:=a])\quad \dfrac{\forall x.(p(x)\supset q(x)),q(b)\vdash q(b)}{\forall x.(p(x)\supset q(x)),q(b)\vdash \exists x.q(x)}\,(\exists\mathrm{right}[x:=b])}{p(a)\vee q(b),\forall x.(p(x)\supset q(x))\vdash \exists x.q(x)}\,(\vee\mathrm{left})}{(p(a)\vee q(b))\wedge\forall x.(p(x)\supset q(x))\vdash \exists x.q(x)}\,(\wedge\mathrm{left})}{\vdash (p(a)\vee q(b))\wedge\forall x.(p(x)\supset q(x))\supset \exists x.q(x)}\,(\supset\,\mathrm{right})$$

図 1: ϕ の証明図

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{P(a)\vdash P(a)\quad \dfrac{Q(a)\vdash Q(a)}{Q(a)\vdash \exists x.Q(x)}\,(\exists\mathrm{right}[x:=a])}{P(a),P(a)\supset Q(a)\vdash \exists x.Q(x)}\,(\supset\mathrm{left})}{(*)P(a),\forall x.(P(x)\supset Q(x))\vdash \exists x.Q(x)}\,(\forall\mathrm{left}[x:=a])\quad \dfrac{Q(b)\vdash Q(b)}{Q(b)\vdash \exists x.Q(x)}\,(\exists\mathrm{right}[x:=a])}{P(a)\vee Q(b),\forall x.(P(x)\supset Q(x))\vdash \exists x.Q(x)}\,(\vee\mathrm{left})}{(P(a)\vee Q(b))\wedge\forall x.(P(x)\supset Q(x))\vdash \exists x.Q(x)}\,(\wedge\mathrm{left})}{\vdash (P(a)\vee Q(b))\wedge\forall x.(P(x)\supset Q(x))\supset \exists x.Q(x)}\,(\supset\,\mathrm{right})$$

図 2: スキーマ Φ の証明図

5 証明システム構成と実行例

5.1 システムの概要

この証明支援システムを $Java^{TM}$ を用いて開発し，現在，LK, LJ, NK*, NJ*と S4 証明システムを実現している．各論理体系は独立にスキーマベース部と，マッチング部からなる類推機構と対話型証明部を備えている．**スキーマベース部:** 各論理体系に対してそれぞれのスキーマとその証明スキーマを対にして蓄積管理する．類推証明ではスキーマの適用可能性を判定するたために，スキーマベースの探索を行うが，スキーマベースを階層化して探索の効率化を図っている [12]. **スキーママッチング部:** 入力された論理式 ϕ に対し，スキーマを探索し，2 階マッチングを用いてマッチング可能かどうかを検査する．マッチング可能なスキーマがあれば，$\Phi\theta = \phi$ なる代入 θ を抽出する．**証明部:**自動証明と対話証明によりなる．

システムを起動すると，証明したい論理式の入力と証明したい論理体系の選択画面が現れる．次に，[手動] または [自動] の選択画面が現れ，そのどちらかをを選んで証明する．証明が完了すると論理体系選択部に戻ることができる．

5.2 対話型証明

対話型証明を選んだ場合，システムはその論理式のシークェントを表示して，証明を開始する．

シークェント中の部分論理式は，すべてボタンで表されている．LK，LJ と S4 においては，シークェント証明を適用したい部分論理式をマウスでクリックすることによって，対応する推論規則が選択され，適用される．選択される推論規則は，このとき選択された論理式をルートとする推論規則となる．その指示に従って適切な推論規則をシークェントに適用し，その結果を表示する．

それに対して，NK*，NJ*と *cut* 規則では，適用する推論規則の情報は特定するのが難しいため，直接推論規則を選ぶように表示ボタンの配置を工夫した．ただし，公理に対してはさらに推論規則を適用することはできないようになっている．また，証明図中の公理でない論理式ならば，どれでもユーザは選択し，推論規則を適用することができる．このため，間違えたときの証明のやり直しや，別解を探すなど，試行錯誤的証明がしやすくなっている．

ユーザは同じ論理式に対して証明体系を切り換えて証明することもできる．異なる体系の証明の違いを知りたい時，4 つのサブシステムを同時に用いて証明することもできる．

5.3 類推証明

類推証明を選んだ場合，入力された論理式をスキーママッチング部に送り，適切なスキーマがあればスキーマベースより取り出す．最後に，その証明スキーマにマッチング代入を適用した証明図を証明実行部で処理

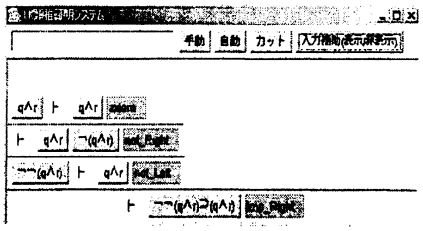

図 5: スキーマ誘導類推証明

をさせ表示する．また，類推処理を行う時，システムはスキーマベースの探索を行うが，スキーマの利用可能性を判定すたために，繰り返しスキーママッチングを行う必要がある．そこで，この回数を減らすことにより，探索を効率化を図ることができる．このため，スキーマベースを部分ベースに分割し階層化することによって，早期に援用できるスキーマを絞り込む手法を用いる [15]．具体的な類推証明の手順は例 9 のようである．

例 9 類推証明

①まず証明したい論理式 $(\neg\neg(q \wedge r) \supset (q \wedge r))$ を図 5 の入力ボックスに入力し，[自動] ボタン (図 5) を押して，類推証明を選ぶ．

②スキーマベースからマッチング可能なスキーマ $(\neg\neg A \supset A)$ が見つかるので，それを用いて論理式の証明を表示する (図 5)．

6 考察

本研究では，一般的な論理体系のためのシークェント計算に基づく証明支援システムを作成した．この類推を実際の論理学の教育支援や教授支援システムとして応用するため，論理式入力支援機構，論理規則適用のための支援機能及び証明図の表示法，といったユーザインターフェースについても工夫した．このシステムは，原理的に自動証明が困難である 1 階述語論理において，大学の講義に用いるには充分な程度の自動証明が可能となった．

しかし，より実用的にするには，さらに非標準論理体系を整備したり，現在の命題様相論理の述語への拡張などがある．また，類推証明も入力された論理式に直接適用可能なスキーマがなくとも，証明途中の部分論理式にスキーマを適用して証明するスキーマの補題的用法による証明を取り入れたシステムへの拡張も考えられる．更に，現在実現しているシステムを数学問題解決支援システムなどに応用するには，本システムに帰納法のスキーマを導入して，帰納法 [3, 7] による証明を可能にすることがある．

参考文献

[1] A.S.Troelstra and H.Schwichtenberg, Basic Proof Theory, Cambridge University Press, 1996.

[2] Cant, A. and Ozols, M.A.: *Xisabelle: A Graphical User Interface to the Isabelle*, Electronics and Surveillance Research Laboratory, Defence Science and Technology Organisation, 1995.

[3] Erica Melis and Jon Whittle, Analogy in Inductive Theorem Proving, 1999.

[4] Huet, G.P. and Lang, B.: *Proving and applying program transformations expressed with second-order patterns*, Acta Informatica **11**, 31–55, 1978.

[5] Harao, M.: *Proof discovery in LK system by Analogy*, Proc. 3rd Asian Computing Science Conference, LNCS **1345**, 197–211, 1997.

[6] 久保 憲吾, 山田敬三, 平田耕一, 原尾政輝, Pre-Checking に基づく効率的スキーママッチングアルゴリズム, 電子情報通信学会 **Vol.J85-D-I**, No.2, pp.143–151, 2002.

[7] Kolbe,T. and C. Walther Adaptaion of Proofs for Reuse, In:D.W.Aha and A. Ram (eds):Adaptation of Knowledge for Reuse. Papers from the 1995 AAAI Fall Symposium, Cambridge, MA, USA.pp.61-67, The AAAI Press, 1995.

[8] Paulson, L.C.: *Introduction to Isabelle*, Computer Laboratory, University of Cambridge, 1995.

[9] Paulson, L.C.: *The Isabelle Reference Manual*, Computer Laboratory, University of Cambridge, 1995.

[10] Paulson, L.C.: *Isabelle's Object-Logics*, Computer Laboratory, University of Cambridge, 1995.

[11] Troelstra, A. S. and Schwichtenberg, H.: *Basec Proof Theory*, Cambridge University Press, 1996.

[12] 山田 敬三, 平田 耕一, 原尾 政輝, "類推機能をもった対話型シークェント計算証明システムの開発", 電子情報通信学会技術報告, AI2003-3, pp 13–16, 2003.

[13] 山田敬三, 平田耕一, 原尾政輝, "スキーママッチングとその計算量", 電子情報通信学会 **J82-D-I**, 11, 1307–1316, 1999.

[14] 尹 淑萍, 山田 敬三, 平田 耕一, 原尾 政輝, シークェント計算における証明支援システムの開発, 火の国シンポジウム 2003 予稿集 216–223, 2003.

[15] 山田敬三, 尹淑萍, 平田耕一, 原尾政輝, 人間指向型類推証明システムにおける類推処理の効率化, 火の国シンポジウム 2004 予稿集 B-9-4, 2004.