社団法人　情報処理学会　研究報告
IPSJ SIG Technical Reports

2005－ＩＣＳ－139（6）
2005／3／14

CF 帰納法における一般化に関する考察—第2報—

山本　泰生[†,††]　井上　克巳[††]

† 神戸大学大学院自然科学研究科　〒658-8501 神戸市灘区六甲台町 1-1
†† 国立情報学研究所　〒101-8430 東京都千代田区一ツ橋 2-1-2
E-mail: †034t257n@y03.kobe-u.ac.jp, ††ki@nii.ac.jp

あらまし　本稿では，帰納論理プログラミングにおける逆伴意法による仮説発見手続きの一手法である CF 帰納法の一般化手続きについて考察する．CF 帰納法は一般節理論の仮説発見に対して健全かつ完全な手続きであるが，その理論面での優位性から現在応用問題への適用が期待されている．一方，構成手続き中の一般化手続きについては，今のところ具体的な操作の紹介があるのみでアルゴリズムとして組織的に構成されるまでには至っていない．そこで本稿では，第1報 [10] で明らかにしたすべての一般化手続きに対して当てはめることのできる順序に関する性質を利用することで，今まで紹介されてきた各一般化操作を再構築し，CF 帰納法における健全かつ完全な新しい一般化手続きを提案する．

キーワード　帰納論理プログラミング，逆伴意法，一般化

Consideration on a generalization procedure in CF-induction —Second Report—

Yoshitaka YAMAMOTO[†,††] and Katsumi INOUE[††]

† Graduate School of Science Techology, Kobe University　1-1 Rokkoudai-cho, Nada-ku, Kobe, 658-8501 Japan
†† National Institute of Informatics　2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, 101-8430 Japan
E-mail: †034t257n@y03.kobe-u.ac.jp, ††ki@nii.ac.jp

Abstract　This paper considers a generalization in CF-induction which is one of procedures for inverse entailment constructing inductive hypothesis in inductive logic programming. CF-induction guarantees soundness and completeness for finding hypotheses from full clausal theories. although some operations have been introduced as generalization procedures in CF-induction, they are not sufficiently enough for the completeness, and a systematic method of constructing hypothese with those generalization operations has not yet been discussed. In this paper, we utilize basic properties that are applicable to any generalization procedure in terms of sequences of operations, and reconstruct a method of hypotheses generation by reorganzing the generalization operators that have been introduced by now. The resulting generalization method is sound and complete for finding hypotheses from full clausal theories in CF-induction.

Key words　inductive logic programming, inverse entailment, generalization

1. はじめに

人工知能分野において，機械学習は中心的な主題であるが，**帰納論理プログラミング** (*Inductive Logic Programming*, 以下 ILP と略す) の研究はこの機械学習と論理プログラミングの共通領域の分野に根ざしている [7]．学習モデルの中でも ILP では，知識を一階述語論理形式により表現しており，入力として**事例** (*example*) とは区別される**背景知識** (*background knowledge*) を新たに導入することができる．そのため ILP は知識表現が複雑な問題に対しても適用可能で，物事を抽象化し整理する能力を扱う概念学習 (*concept learning*) だけでなく，ニューラルネットワークや決定木では行うことが困難な，欠落した理論を発見する理論補完 (*theory completion*) にも利用することができる．

入力として背景知識 B と観測事例 E が与えられたとき，次の2式を満足するような仮説 H を出力することが ILP のタスクとなる．すなわち仮説 H は背景知識 B と無矛盾で，B とと

もに観測事例 E を論理的に帰結する.

$$B \wedge H \models E \tag{1}$$

$$B \wedge H \text{ が無矛盾} \tag{2}$$

ILP の枠組みでこれまでさまざまな手法が提案されているが, 最もよく知られている手法の一つは**逆伴意法** (*inverse entailment*) [6] と呼ばれる手法である. 逆伴意法では, 式 (1) を等価変換した次式を用いる.

$$B \wedge \neg E \models \neg H \tag{3}$$

式 (3) は入力データである B と E の否定から出力される仮説 H の否定が演繹計算 (*deductive calculation*) によって求められることを意味している. ここで演繹計算のみで直接仮説の否定を求めるようなことはせず, 次式を満足するような**橋渡し理論** P(*brige formula*) を生成する.

$$B \wedge \neg E \models P \models \neg H \tag{4}$$

式 (4) に基づき逆伴意法は次の 2 つの手続きから構成される.
手続き (1): $B \wedge \neg E$ から次式を満足する橋渡し理論 P を構成する.

$$B \wedge \neg E \models P \tag{5}$$

手続き (2): 手続き (1) で構成した P より, 次式を満足する仮説 H を生成する.

$$P \models \neg H \tag{6}$$

本稿では手続き (2) のことを**一般化** (*generalization*) 手続きと呼ぶ.

実世界で数多くの応用事例を持ち ILP の中で最もよく知られる手法の一つである Progol [6] も逆伴意法を用いている. Progol では橋渡し理論を $B \wedge \neg E$ より帰結されるすべての単位節の連言で構成する. 橋渡し理論を構成後, 橋渡し理論からの一般化手続きは式 (6) を等価変換した次式に基づく帰納計算 (*inductive calculation*) により構成される.

$$H \models \neg P \tag{7}$$

式 (7) を用いた一般化手続きの場合, 橋渡し理論からその否定式を求める必要があるが, Progol ではこの橋渡し理論の否定はボトム節と呼ばれる単一の節となる. しかしながら, 橋渡し理論としてボトム節を利用したとき, 式 (7) を満たすどのような帰納計算を行っても生成することができない仮説が存在する. すなわち Progol は仮説発見に対して健全ではあるが完全でない手続きである. また入力される背景知識 B, 観測事例 E についてもホーン節理論でなければならないという制約がある.

その中で本稿が取り挙げているのは, 一般節理論の仮説発見に対して健全かつ完全な手続きである **CF 帰納法** (*CF-induction*) [4] である. 文献 [4] に依れば, 橋渡し理論として $B \wedge \neg E$ から帰結される定理のうち**特徴節** (*characteristic clause*) に注目し特徴節から構成される節理論を橋渡し理論として選択すると, 式 (7) を満足するすべての帰納計算を行うことで一般節理論の仮説を完全に生成することができる. 近年, 一般節理論の仮説発見に対し完全性を保証するという理論的優位性から分子生物学などのさまざまな応用問題に CF 帰納法を適用しようとする動きが高まりつつある. しかしながらこれまでの CF 帰納法では逆伴意法の手続き (1) の橋渡し理論の構成に関しては厳密な手続きが用意されている一方で, 橋渡し理論から仮説を生成する一般化手続きに関してはいくつかの ILP 技法が紹介されているのみで, アルゴリズムとして組織的に構成されるまでには至っていない.

橋渡し理論から仮説を生成する一般化手続きは, 式 (6), (7) のどちらの式を用いるにせよ伴意関係を利用して計算することになる. 第 1 報 [10] では, この伴意に基づく導出手続きに関して, Subsumption, Weakening, Resolution の各基本操作の適用順序が一意に固定できることを明らかにした. そこで本論文では, この性質を式 (6) に適用し, CF 帰納法における橋渡し理論と仮説の否定との関係を考察する. その中で, 変数が出現しない仮説は橋渡し理論から節の挿入操作のみで健全かつ完全に生成できることを示す. 提案手法は, 変数出現も含む任意の仮説発見に対して健全かつ完全な手続きで, この節の挿入操作を用いている. すなわち, はじめに変数が出現しない仮説候補の否定を橋渡し理論から節の挿入操作によって生成し, DNF–CNF 変換より仮説候補を求める. その後必要ならば逆代入操作より変数を出現させる.

本論文は, 次のように構成されている. 2 章で本稿を理解する上で必要となる用語と CF 帰納法において重要な概念である特徴節について説明し, 3 章で CF 帰納法ついて述べる. 4 章では一般化について考察し, 最後に 5 章でまとめる.

2. 準　　備

この章では, 本稿を理解する上で必要となる基礎的な用語と CF 帰納法を理解する上で必要となる特徴節について説明する.

2.1 帰納論理プログラミング

原子文 (アトム)(*atom*) とは n 引数の関係定数 と n 個の項を組み合わせたものである. すなわち, p を述語記号, t_i を項としたとき $p(t_1, \ldots, t_n)$ をアトムという.

リテラル (*Literal*) とは原子文またはその否定である. 原子文のことを**正リテラル**と呼び, 原子文の否定のことを**負リテラル**と呼ぶ.

節 (*clause*) は重複したリテラルを持たないリテラルの選言であり, 各節はしばしばリテラルの集合を用いて表される. 節中のすべての変数は, 先頭で全称限量化されていると仮定する. 節の**長さ** (*length*) とは節が持つリテラルの数である. **空節** (*empty clause*) は長さ 0 の節であり, □ で表す. また, 特に長さが 1 の節を**単位節** (*unit clause*) という. **正 (負) 節** (*positive (negative) clause*) はその節中のリテラルがすべて正 (負) リテラルである節である. **ホーン節** (*Horn clause*) は確定節と負節のことである. その他の節は**非ホーン節** (*non-Horn clause*) という. 節理論は非ホーン節を含んでいれば**一般** (*full*) 節理論という.

連言標準形 (CNF)(*conjunctive normal form*) 式は節の連言であり，**選言標準形 (DNF)**(*disjunctive normal form*) 式はリテラルの連言の選言である．節理論 Σ は Σ 中のすべての節の連言である CNF と同一視する．節理論 $\Sigma = C_1 \wedge \cdots \wedge C_k$(ただし各 C_i は節) の補を DNF 式 $\neg C_1\sigma_1 \vee \cdots \vee \neg C_k\sigma_k$ として定義する．ただし，$C_i = (B_1 \wedge \cdots \wedge B_m \supset A_1 \vee \cdots \vee A_m)$ に対して $\neg C_i = B_1 \wedge \cdots \wedge B_n \wedge \neg A_1 \wedge \cdots \wedge \neg A_m$ であり，σ_i は C_i 中の各変数 x をスコーレム定数 sk_x に置き換えたものである．この変数の置き換えは $\neg C_i$ 中の各変数が先頭で存在限量化されていることを反映している．あいまいさがないので Σ の補集合を $\neg\Sigma$ とかく．

Σ_1，Σ_2 を節理論とする．Σ_1 が真のとき，必ず Σ_2 も真となるような関係を**伴意** (*entailment*) **関係**と呼ぶ．また $\Sigma \models C$ のとき，節 C は節集合 Σ の **(論理的) 帰結** (*consequence*) または**定理**であるという． Σ の定理であるすべての節の集合を $Th(\Sigma)$ で表す．

C と D を２つの節とする．$C\theta \subseteq D$ となるような代入 θ が存在すれば C は D を**包摂する** (*subsume*) といい，$C \succeq D$ と書く．C は D を包摂するが D は C を包摂しないとき C は D を**真に包摂する** (*properly subsume*) という．節理論 Σ に対して，$\mu\Sigma$ は Σ 中のすべての節により真に包摂されない Σ 中の節の組によりあらわされる．また節理論 Σ_1 が節理論 Σ_2 を包摂するとは，Σ_2 中に任意の節 C について Σ_1 のある節 C' が存在し $C' \succeq C$ となることを意味する．

2.2 特徴節

特徴節は CF 帰納法の手続きを説明する上で重要な概念である．はじめに，ある表現言語の語彙で構築される**生成領域** (*production field*) なるものを定義する．

［定義 2.1］ **生成領域**

(1) $\mathcal{R}$ を言語におけるすべての述語記号とする．$\mathbf{R} \subseteq \mathcal{R}$ に対して，すべての $\mathbf{R}$ からの正 (負) の出現の集合を $\mathbf{R}^+(\mathbf{R}^-)$ で表す．リテラルの集合 $\mathcal{R}^+ \cup \mathcal{R}^-$ は $\mathcal{L}$ と表される．変数を持つリテラルが $\mathbf{L}$ に存在するときはいつもそのすべての例も $\mathbf{L}$ に存在するならば，リテラルの集合 $\mathbf{L} \subseteq \mathcal{L}$ は**例化** (*instantiation*) **のもとで閉じている**という．

(2) 生成領域 $\mathcal{P}$ は $\langle \mathbf{L}, Cond\rangle$ で与えられる．ここで，$\mathbf{L}$ を $\mathcal{P}$ の**特徴リテラル**といい，$\mathcal{L}$ の部分集合であり，例化のもとで閉じているものとする．$Cond$ は満たされるべき条件で，$Cond$ が指定されない場合，生成領域 $\mathcal{P}$ は $\langle \mathbf{L}\rangle$ と記述される．

(3) ある節 C のリテラルがすべて $\mathbf{L}$ に属しかつ C が $Cond$ を満たすならば，節 C は生成領域 $\mathcal{P}=\langle \mathbf{L}, Cond\rangle$ に**属する**という．Σ を節集合とするとき，$Th_{\mathcal{P}}(\Sigma)$ は Σ の定理であり $\mathcal{P}$ に属するすべての節の集合を表す．

(4) 節 D が $\mathcal{P}$ に属しており，C が D を包摂するときはいつも C も $\mathcal{P}$ に属するならば，生成領域 $\mathcal{P}$ は**安定している**という

例えば，安定した生成領域の条件 $Cond$ には，節の長さ，特徴リテラルに関しては正または負リテラルの選択つきの述語名，等を指定することができる．

生成領域を定義することで**特徴節** (*characteristic clauses*) なる与えられた問題に関する「面白い」節を表現することができる．

［定義 2.2］ **特徴節**

Σ を節集合，$\mathcal{P}$ を安定した生成領域とするとき，$\mathcal{P}$ に関する Σ の**特徴節**の集合を次で定義する．

$$Carc(\Sigma, \mathcal{P}) = \mu Th_{\mathcal{P}}(\Sigma)$$

新しい節 F が節集合 Σ に追加されたときには，この新たな情報によりいくつかの帰結を得る．この新たに導かれた新しい"面白い"節は**新特徴節** (*new characteristic clauses*) と呼ばれ，次の式で定義される．

［定義 2.3］ **新特徴節**

Σ を節集合，$\mathcal{P}$ を安定した生成領域，F を式とするとき，Σ と $\mathcal{P}$ に関する F の新特徴節の集合を次で定義する．

$$Newcarc(\Sigma, F, \mathcal{P}) = \mu[Th_{\mathcal{P}}(\Sigma \cup \{F\}) - Th_{\mathcal{P}}(\Sigma)]$$

3. CF 帰納法

節理論として背景知識 B，観測事例 E が与えられたとき，逆伴意法を用いる多くの手法では (3) を用いて直接的に仮説の否定を求めることせず，式 (4) を満たすような橋渡し理論 P を導入している．

橋渡し理論 P は生成条件として式 (5) の他に，$B \wedge H$ が無矛盾であるという制約も含まれる．$B \wedge H$ が無矛盾であることは，

$$B \not\models \neg H \tag{8}$$

と表せる．もし仮に B だけから P が帰結されれば，式 (6) より，B から $\neg H$ が帰結されるので式 (8) に矛盾する．よって，

$$B \not\models P \tag{9}$$

以上から逆伴意法で生成される橋渡し理論 P は式 (5) と (9) を満たさなければならない．

CF 帰納法は，Inoue [4] により提案された仮説発見のための手法であるが，CF 帰納法における橋渡しの理論は (以後，CF 帰納法の橋渡し理論を CC と呼ぶ．)，仮説の否定の表現語彙を特徴リテラルとする生成領域 $\mathcal{P}$ に関する**特徴節**によって構成される．その構成法は次の通りである．

［定義 3.1］ $CC = C_1 \wedge \ldots \wedge C_m$ について，各 C_i が次の条件を満足する節のとき CC を CF 帰納法における橋渡し理論と呼ぶ．

（1） 各 C_i は $Carc(B \wedge \neg E, \mathcal{P})$ 中の節である．

（2） 少なくとも C_i のひとつは $NewCarc(B, \neg E, \mathcal{P})$ の節である．

構成されるすべての節が $B \wedge \neg E$ の特徴節なので，CC は (5) を満たす．また CC 中の少なくとも一つの節は $\neg E$ に関する新特徴節なので B のみから帰結されることはなく式 (9) も満たす．

さらに $B \wedge \neg E$ から帰結されるどのような節理論も，ある橋渡し理論 CC が存在し，CC から帰結される．すなわち任意の仮説 H はある CC と次の伴意関係を持っている．

$$CC \models \neg H \tag{10}$$

このことから，橋渡し理論から式 (10) に従い生成されうるすべての仮説を求めると，CF 帰納法は，仮説発見に対して健全かつ完全な手続きとなる．橋渡し理論 CC から H を生成する一般化手続きについて，文献[4] では，式 (10) を等価変換した次式を利用している．

$$H \models \neg CC \tag{11}$$

すなわち $\neg CC$ を CNF 式に変換したのち，帰納計算によって仮説を生成する．帰納計算を実現する ILP 技法として，現在以下のような手法がある．

- 逆スコーレム化 (*Reverse Skolemization*)： スコーレム定数及びスコーレム関数を変数に置き換える．
- 反例化 (*Anti-instantiation*): 基礎項を変数に置き換える．
- リテラル除去 (*Anti-subsumption*)： ある節からいくつかリテラルを除去する．
- 節理論追加 (*Anti-weakening*) :節理論を連言で追加する．
- 逆融合 (*Inverse resolution*)： 導出原理の逆の操作を行う．この操作を行うと式 F 中には出現しない新しい述語名やリテラルを獲得することができる．
- 最小汎化 (*Least general generalization*)： 2 つの項の異なる部分を変数化して，式の共通パターンを得る．

4. 一　般　化

従来の CF 帰納法では，橋渡し理論に関する厳密な構成手続きがあるものの橋渡し理論から仮説を生成する一般化手続きに関しては 3 章で挙げたような ILP 技法が紹介されているのみで，これらをアルゴリズムとして組織的に構成するまでには至っていなかった．橋渡し理論に対し上の操作はどれを何度適用しても仮説となる．逆に CF 帰納法が仮説発見に対して完全となるためには，行うことのできるすべての操作と選択を考える必要がある．本章では，この一般化手続きに関して従来のさまざまな ILP 技法を再構築し，完全な一般化手続きを構成する基本操作を明らかにする．

橋渡し理論 CC を入力節理論と考えると，一般化手続きとは，次式を満たすような節理論 H を生成する手続きのことといえる．

$$CC \models \neg H \tag{12}$$

$$H \models \neg CC \tag{13}$$

式 (12) を利用する一般化手続きでは，演繹計算を用いて CC から仮説の否定を生成し，その後，CNF–DNF 変換を行い仮説を生成する手順となる．一方，式 (13) を利用する一般化手続きでは，はじめに CNF–DNF 変換を行い $\neg CC$ の CNF 式を求め，その後帰納計算を用いて仮説を生成する．そのどちらを利用するにせよ一般化手続きは伴意関係に基づく計算手続きなので，はじめに節理論における伴意関係について考察する．

4.1 導出手続き

次式 (14) が成り立つ S，T において，式 (14) を満たす任意の T は，S に対して次に示す 3 つの基本的な導出手続きを順次適当に行うことで求められる．[9]

$$S \models T \tag{14}$$

（1） (Weakening) $T = S - \{C\}$，ただし $C \in T$．

（2） (Resolution) $T = S \cup \{R\}$，ただし R は，節 C，D (C, $D \in S$) の融合節．

（3） (Subsumption) $T = S \cup \{D\}$，ただし D は，$C \succeq D$ ($C \in S$)

［定義 4.1］ **導出手続き (Weakening)**
節理論 S に対し，Weakening を一度行って得られる節理論を T とするとき，$S \vdash_W T$ と表記する．また Weakening の操作を複数回 (1 度も含める) 行って得られる場合は，$\vdash_W^*$ の記号を用いる．

［定義 4.2］ **導出手続き (Resolution)**
節理論 S に対し，Resolution を一度行って得られる節理論を T とするとき，$S \vdash_R T$ と表記する．また Resolution の操作を複数回 (1 度も含める) 行って得られる場合は，$\vdash_R^*$ の記号を用いる．

［定義 4.3］ **導出手続き (Subsumption)**
節理論 S に対し，Subsumption を一度行って得られる節理論を T とするとき，$S \vdash_S T$ と表記する．また Subsumption の操作を複数回 (1 度も含める) 行って得られる場合は，$\vdash_S^*$ の記号を用いる．

［例 4.1］

$$\begin{aligned} S &= (p(a) \vee q(X,a)) \wedge \neg p(a) \\ T &= q(b,a) \end{aligned}$$

$S \models T$ であるが，実際，

$$\begin{aligned} S &\vdash_R (p(a) \vee q(X,a)) \wedge \neg p(a) \wedge q(X,a) \\ &\vdash_S (p(a) \vee q(X,a)) \wedge \neg p(a) \wedge q(X,a) \wedge q(b,a) \\ &\vdash_W^* q(b,a) = T \end{aligned}$$

ところで上の定義による伴意に基づく導出手続きはスコーレム定数を出現させる手続きを含まないので S，T にスコーレム定数が出現するとき，$S \models T$ が成り立つにも関わらず 3 操作より S から T が求められない場合がある．

［例 4.2］

$$\begin{aligned} S &= \neg p(a) \vee q(a) \\ T &= \neg p(sk_X) \vee q(sk_X) \end{aligned}$$

ただし sk_x はスコーレム定数．
$S \models T$ であるが，実際，3 つの基本操作からでは S から T は生成されない．スコーレム定数が出現しない場合，伴意に基づく導出手続きは 3 つの基本操作より完全に構成できるので，以後 S，T にスコーレム定数が出現しない場合を考察する．

さて第一報では，これら導出手続きの適用順序に関して次の定理を明らかにしている [10]．

［定理 4.1］ $S \models T$ を満たす任意の一般節理論 S，T は次式のように表わせる．ただし U，V は節理論．

$$S \vdash_R^* U \vdash_S^* V \vdash_W^* T \tag{15}$$

次に導出手続きとして下の 2 操作を考える.

（1） (代入) $T = S \cup \{C\theta\}$, ただし $C \in S$, θ は代入.

（2） (節の挿入) $T = \{S - \{C\}\} \cup \{C \vee M\}$, ただし $C \in S$, M は任意の節.

［定義 4.4］ **導出手続き (代入)**
節理論 S に対し, 代入操作を一度行って得られる節理論を T とするとき, $S \vdash_{S_{sub}} T$ と表記する. また代入の操作を複数回 (1 度も含める) 行って得られる場合は, $\vdash^*_{S_{sub}}$ の記号を用いる.

［定義 4.5］ **導出手続き (節の挿入)**
節理論 S に対し, 節の挿入操作を一度行って得られる節理論を T とするとき, $S \vdash_{S_{ins}} T$ と表記する. また節の挿入操作を複数回 (1 度も含める) 行って得られる場合は, $\vdash^*_{S_{ins}}$ の記号を用いる.

代入, 節の挿入操作とも Subsumption に含まれる操作であるが, 実はこの 2 操作が任意の Subsumption 手続きを構成する基本操作となる [11].

［**定理** 4.2］ $S \vdash^*_S T$ を満たす任意の節理論 S, T について, ある節理論 U が存在して, 次式のように表わせる.

$$S \vdash^*_{S_{sub}} U \vdash^*_{S_{ins}} T \tag{16}$$

4.2 橋渡し理論と仮説の関係

4.1 章を踏まえてここでは CF 帰納法における橋渡し理論 CC と仮説の否定 $\neg H$ の関係を考察する. はじめ次式 (12) に注目する.

$$CC \models \neg H$$

上式に定理 4.1 を順次適用すると, 次式のように表わせる. (ただし, U, V, W はある節理論)

$$CC \vdash^*_R U \vdash^*_S V \vdash^*_W \neg H \tag{17}$$

ここで注意しなければならないのが例 4.2 にある通り, スコーレム定数が出現するような $\neg H$ は式 (17) より生成することができない点である. $\neg H$ にスコーレム定数が出現するのは H に変数が出現する場合に限るので, 式 (17) を用いて橋渡し理論 CC から $\neg H$ を生成する導出手続きは, 変数が出現しないような H に限定した場合完全性は保証される. 4.2 章では, 以後変数が出現しない仮説に限定して考える.

式 (17) は CF 帰納法の橋渡し理論 CC の選択方法の定義よりさらにシンプルな表現となる [11].

［**定理** 4.3］ ある橋渡し理論 CC から次式

$$CC \vdash_R V$$

より導出できるすべての節理論 V は, 別の橋渡し理論 CC' を用いて次式

$$CC' \vdash^*_{S/W} V$$

より求めることができる.

この定理は橋渡し理論としてすべての選択方法を考える場合, Resolution 操作を Subsumption 操作と Weakening 操作によって置き換えられることを意味する. この定理を利用することにより, 比較的計算コストの高い Resolution 操作を用いることなく橋渡し理論から仮説の否定を生成することができる. 定理 4.3 を式 (17) へ適用すると次の系 4.1 を得る [11].

［**系** 4.1］ 選択可能なすべての橋渡し理論に対し,

$$CC \vdash^*_S U \vdash^*_W \neg H \tag{18}$$

を満たす導出手続きをすべて適用すれば $\neg H$ は健全かつ完全に生成できる.

また式 (18) に定理 4.2 を適用すると, ある節理論 V が存在し, Subsumption 操作を代入操作と節の挿入操作に分解した次式のように書き直せる.

$$CC \vdash^*_{S_{sub}} V \vdash^*_{S_{ins}} U \vdash^*_W \neg H \tag{19}$$

ここで注意しなければならないのは, 定理 4.4 から得られた上式 (19) に基づく導出手続きは完全であることは保証しているが健全であることはまだ何も述べていない点である. だが実際は, 式 (19) に基づく導出手続きにより得られる $\neg H$ は, 式 (18) に基づく導出手続きによっても生成することができ健全である.

また代入操作についても定理 4.3 に似た次の定理 4.4 が成り立つ [11].

［**定理** 4.4］ V を節理論とすると, 次式

$$CC \vdash_{S_{sub}} V$$

が成り立つとき, V と同一の橋渡し理論を構成することができる.

式 (19) に定理 4.4 を適用すると, 次の系 4.2 が成立する [11].

［**系** 4.2］ 選択可能なすべての橋渡し理論に対し,

$$CC \vdash^*_{S_{ins}} U \vdash^*_W \neg H \tag{20}$$

を満たす導出手続きをすべて適用すれば $\neg H$ は完全に生成できる.

さらに節の挿入操作と Weakening 操作からなる導出手続きについても次の定理 4.5 が成り立つ [11].

［**定理** 4.5］ U を節理論とする. ある橋渡し理論 CC から次式の導出手続きより節理論 T が求まるとする.

$$CC \vdash^*_{S_{ins}} U \vdash_W T$$

このとき,

$$CC' \vdash^*_{S_{ins}} T$$

を満足するような別の橋渡し理論 CC' が存在する.

定理 4.5 を式 (20) に, 適用すると次の系 4.3 が得られる [11].

［**系** 4.3］ 選択可能なすべての橋渡し理論に対し,

$$CC \vdash^*_{S_{ins}} \neg H \tag{21}$$

を満たす導出手続きをすべて適用すれば $\neg H$ は完全に生成できる.

4.3 提 案 手 法

4.2 章より, 変数が出現しない, すなわち出現が基礎項のみである任意の仮説は, 考えられうるすべての橋渡し理論より式 (21) に従い, 節の挿入操作を適用後, 得られた節理論の否定の式から DNF–CNF 変換を行うことで生成することができる.

ここで変数を含むすべての仮説を生成する手順を考える．変数を含む仮説 H について，$H \models \neg CC$ を満たす，出現が基礎項のみの橋渡し理論 CC が存在する [4]．よって H のエルブラン基底の部分集合で，$\neg CC$ を帰結するような有限集合 S が存在する．このとき S は基礎項のみしか出現しない仮説なので，S は上のアルゴリズムより生成することができる．また S は H に出現する変数を基礎項に置き換えた節より構成されているので S に対して逆代入操作を行うことで H を求めることができる．また節挿入操作と逆代入操作はそれぞれ健全な操作である．

以上から，橋渡し理論 CC から仮説 H を求める健全かつ完全な一般化手続きは，次のように構成される．

健全かつ完全な仮説発見手続き

入力: 橋渡し理論 CC

出力: 仮説 H

（1） CC に対して節の挿入操作を何度か (0 回も含む) 適用する．

（2） 1 で生成された節理論の否定を CNF 式に変換する．

（3） 2 で生成した節理論に対して逆代入操作を何度か (0 回も含む) 適用する．

（4） 3 で生成した節理論が仮説である．

次に提案手法を用いて仮説発見を行う例を示す．

[例 4.3]

$$B_1 = even(0) \land (\neg odd(x) \lor even(s(x)))$$
$$E_1 = even(s^3(0))$$
$$Target : (\neg even(0) \lor odd(s(0))) \land (\neg even(s^2(0)) \lor odd(s^3(0)))$$

ここで $s^1(0) = s(0)$ で，$n > 1$ については $s^n(0) = s(s^{n-1}(0))$ とする．次のように橋渡し理論 CC を構成する．

$$CC = even(0) \land (\neg odd(s(0)) \lor even(s^2(0))) \land \neg odd(s^3(0))$$

ここで $even(0)$ に $even(s^2(0))$，$\neg odd(s^3(0))$ に $\neg odd(s(0))$ をそれぞれ挿入し仮説の否定とする．すなわち，

$$\begin{aligned}\neg H = & (even(0) \lor even(s^2(0))) \\ & \land (\neg odd(s(0)) \lor even(s^2(0))) \\ & \land (\neg odd(s^3(0)) \lor \neg odd(s(0)))\end{aligned}$$

となるので，

$$\begin{aligned}DNF(H) = & (\neg even(0) \land \neg even(s^2(0))) \\ & \lor (odd(s(0) \land \neg even(s^2(0)))) \\ & \lor (odd(s^3(0) \land odd(s(0)))\end{aligned}$$

$$\begin{aligned}CNF(H) = & (\neg even(0) \lor odd(s(0))) \\ & \land (\neg even(s^2(0)) \lor odd(s^3(0)))\end{aligned}$$

となりターゲットの仮説を生成することができる．ただし $DNF(H)$ は DNF 形式で表現された H，$CNF(H)$ は CNF 形式で表現された H をそれぞれ表わす．またこの仮説に逆代入操作を適用することで次の仮説も生成することができる．

$$H = \neg even(X) \lor odd(s(X))$$

この例に関して，従来の一般化手続きでは，上の CC に対して $\neg CC$ の CNF 式

$$\begin{aligned}CNF(\neg CC) = & (\neg even(0) \lor odd(s(0)) \lor odd(s^3(0))) \\ & \land (\neg even(s^2(0)) \lor odd(s^3(0)) \lor \neg even(0))\end{aligned}$$

を計算し，一般化操作として各節で，$odd(s^3(0))$，$\neg even(0)$ をそれぞれリテラル除去することでターゲット仮説を生成することができる．このことから上の節の挿入操作によってリテラル除去操作と同等の処理が行われていることがわかる．

提案手法は，橋渡し理論から演繹計算によって直接的に仮説の否定を生成するため，仮説の否定から仮説を生成する DNF–CNF 変換について，冗長な節を含まないような簡約化された出力を求める効率的手法が必要となる．

5. おわりに

本稿は，CF 帰納法における一般化の手法に関する研究をまとめたものであり，伴意関係に基づく導出手続きの適用順序に関する性質を利用することで，変数を含まない仮説の否定が CF 帰納法における橋渡し理論より節の挿入操作のみで生成できることを示した．さらにこの特徴を用いた仮説発見に対して健全かつ完全な一般化手続きとして，節の挿入操作と逆代入操作のみで構成される新しい手法を提案した．

本論文により，CF 帰納法における健全かつ完全な一般化手続きが 2 操作のみから構成できることが明らかとなり，これまでいくつかあった一般化操作を体系的に整備することができた．今後は提案手法の実装の観点から，有効な仮説の否定を効率的に探索するような戦略的な節挿入操作について考察したい．

文　　献

[1] Cussens, J and Frisch, A. ILP: A Short Look and a Longer Look Forward. *Journal of Machine Learning Research*, 4, 415–430, 2003.

[2] 古川 康一. 帰納論理プログラミング. 共立出版, 2001.

[3] Inoue, K. Induction, abduction, and consequence-finding. *Proceeding of the 11th International Conference on Inductive Logic Programming* (ILP'01), *Lecture Notes in Artificial Intelligence*, 2157, 65–79, 2001.

[4] Inoue, K. Induction as consequence finding. *Machine Learning*, 55, 109–135, 2004.

[5] Inoue, K. Linear resolution for consequence finding. *Artificial Intelligence*, 56, 301–353, 1992.

[6] Muggleton, S. Inverse entailment. *New Generation Computing*, 13, 245–286, 1995.

[7] Nada, L and Sašo, D. Inductive Logic Programming Techniqes and Applications. Ellis Horwood, 1994.

[8] Oliver, R. Hybrid Abduction Induction Learning:a Generalisation of Progol. *Proceeding of the 13th International Conference on Inductive Logic Programming* (ILP'03), *Lecture Notes in Artificial Intelligence*, 2835, 311–328, 2003.

[9] Yamamoto, A. Hypothesis Finding based on Upward Refinement of Residue Hypotheses. *Theoretical Computer Science*, 298, 5–19, 2003.

[10] 山本 泰生. CF 帰納法における一般化に関する考察. 電子情報通信学会技術研究報告 (人工知能と知識処理), 103, no.623, 43–48, 2004.

[11] 山本 泰生. CF 帰納法における一般化手続きに関する研究とその実装. 神戸大学大学院自然科学研究科修士論文, 2005.