# コストに基づく仮説推論における最適解探索の一方法

加藤昇平† 世木博久† 伊藤英則†

本論文では，一階述語論理ホーン節で表現された知識ベースを対象にして，与えられた観測を説明する最小コストの仮説集合を求める仮説推論システムを提案する．仮説推論では一般に，観測を説明する仮説集合は複数存在するが，計画・診断問題等の応用では，すべての説明が必要とされるのではなく，最も好ましい説明が要求されることが多い．したがって本論文では，与えられた観測を説明する最小コストの仮説集合を効率的に求めるために，仮説推論の推論過程における探索を $A^*$ アルゴリズムを用いて制御する．さらに，その探索制御能力を決定づけるヒューリスティック評価関数の導出方法も提案する．また，本論文で提案する推論方法を前向き推論を用いて計算機上に実装した実験結果についても報告する．

# Optimal Search in Cost-based Horn Abduction

SHOHEI KATO,† HIROHISA SEKI† and HIDENORI ITOH†

We propose an efficient first-order cost-based Horn abductive reasoning system, which can find a minimal-cost explanation of a given observation. In general, abductive reasoning might find more than one solution. We, however, do not always require all the solutions in certain applications. We often need the most preferable solution instead. In this paper, we introduce a search control technique of A* into abductive reasoning mechanism, thereby obtaining efficiently a minimal-cost explanation of a given observation. We also propose a method to derive an effective (accurate and efficiently computable) admissible heuristic. We then implement our cost-based abductive reasoning system and show some experimental results.

## 1. はじめに

仮説推論は，不完全な知識の下で適切な推論を行う高次推論の一形式であり，足りない知識を仮説で補って推論を進めるという，柔軟で知的な知識処理を行うことができる．したがって，人工知能の実用化の一例として現在広く使われている診断や設計・計画のためのエキスパートシステムの能力向上のための鍵となる推論処理として，多くの研究結果が報告されている（例えば，文献1)，2)）．しかしながら，仮説推論システム実現上の問題点として，仮説推論の計算量が極めて大きいことが知られている[3),4)]ので，いかにして仮説推論の探索空間を絞り込む工夫を行うかということが重要な課題となる．

Pooleによる定式化[1)]においては，仮説推論とは，公理の集合（ホーン節集合）$F$ と観測（ゴール）$O$ が与えられた時，$F \cup E$ から $O$ が説明できて（すなわち，$F \cup E \vdash O$），かつ $F \cup E$ は無矛盾である（すなわち，$F \cup E \not\vdash false$）ような仮説の集合 $E$ を求める問題である．仮説推論をこのような論理的枠組で捉えることにより，論理プログラミングや演繹データベースの分野における様々な問合せ処理技術を用いて仮説推論が効率的に実現されている（例えば，文献5)～7)）．

仮説推論では一般に，観測を説明する仮説集合は複数存在する．しかしながら，計画・診断問題等において見られるように，ユーザが要求する解はすべての説明ではなく，むしろ最も好ましい説明であることが多い．例えば，計画問題においては，与えられた目標を達成するすべての手順よりも，それらの中で最も安価（あるいは最速な）手順がしばしば要求される．そこで文献8)～11)などの研究では，より適切な説明を得るために，仮説に好ましさの基準（確率，コスト等）を与える手法が報告されている．上記のような要求は，与えられた観測をより適切に説明するような仮説をどのように選択するかという「合理的な仮説の選択」の問題として認識されている[2)]．

そこで本論文では，一階述語論理ホーン節で表現された知識ベースを対象に，与えられた観測を説明する最小コストの仮説集合を求めることにより最適な説明

† 名古屋工業大学 知能情報システム学科
Department of Intelligence and Computer Science, Nagoya Institute of Technology

を得る仮説推論について考える☆．各仮説に好ましさの基準が与えられることによって，最適な説明を求める問題は，ヒューリスティック探索の問題に帰着される．したがって本論文では，仮説推論を実現する推論処理技術に対して $A^*$ アルゴリズム[12]の持つ探索制御技術を導入する．さらに，本研究で提案する推論方法の探索制御能力を決定づけるヒューリスティック評価関数の導出方法を提案する．

本論文の構成は以下の通りである．まず 2 章でコストに基づく仮説推論の枠組について述べ，関連研究について言及する．次に 3 章で A* アルゴリズムに基づき最適な説明を求める効率的な推論方法を後向き推論の枠組で説明し，4 章でこの推論方法の前向き推論による実装について述べる．そして，5 章で実験結果について説明する．

## 2. 仮説推論

この節では，Poole による定式化[1]に基づきつつ，本論文で用いる仮説推論の枠組について説明し，この分野での関連研究について述べる．

### 2.1 コストに基づく仮説推論

本論文では，仮説の選択の基準として知識ベースに含まれる各仮説に重み（コスト）が与えられている場合を対象に，与えられた観測に対して最適な説明を求める仮説推論を考える．各仮説に与えられる重みは正の数とし，説明が最適であるとは説明のコスト☆☆が最小であることとする．

**定義 2.1** $F$ をホーン節集合（「事実」と呼ぶ），$H$ を基底単位節の集合（「仮説集合」と呼ぶ）とする．また，$O$ を存在束縛されたアトムの連言（「観測」，または単に「ゴール」と呼ぶ）とする．このとき，$O$ の $F \cup H$ による最適な説明とは，以下の条件を満足するような，$H$ の要素の代入例から成るマルチセット $E$ を求めることである．

$F \cup E \vdash O$ （$F \cup E$ から $O$ が証明される） (AR1)

$F \cup E \not\vdash false$ （$F \cup E$ は無矛盾である） (AR2)

$cost(E) \leq cost(D)$ for all $D$ : AR 1, AR 2
（条件 AR 1, AR 2 を満たすマルチセットの中で $E$ のコストが最小） (AR3)

$cost(H)$ は仮説のマルチセット $H$ に含まれる仮説のコストの和を表すとする．ここで，$F$ は常に成り立つ知識として扱われる．一方で，$H$ の代入例からなる部分集合は $F$ と矛盾する可能性がある．また，$F$ と $H$ の和集合をプログラムと呼び，$P$ で表記する．なお，本論文では説明 $E$ はマルチセットとして扱われる． □

| 事実 | 仮説 | コスト | 仮説 | コスト |
|---|---|---|---|---|
| $p(X,Y) \leftarrow e(X), s(Y).$ | $q(1).$ | 3 | $q(2).$ | 2 |
| $p(X,Y) \leftarrow f(X), t(Y).$ | $r(1).$ | 5 | $r(2).$ | 2 |
| $e(X) \leftarrow q(X).$ | $s(1).$ | 6 | $s(2).$ | 4 |
| $f(X) \leftarrow r(X).$ | $t(1).$ | 3 | $t(2).$ | 2 |
| $false \leftarrow r(X), t(X).$ | | | | |

図 1 例題プログラム $P_{ex}$
Fig. 1 An example $P_{ex}$.

**注 2.1** 本研究で提案するシステムの一応用例である計画問題においては，例えば，観測の説明 $E$ は与えられた目標状態を達成する作業手順として表現される☆☆☆．このような場合には，$E$ をマルチセットとして扱い，$E$ の要素について重複を許したり，順序を考えるほうが好ましいと考えられる．観測の説明を仮説の集合として表現する定式化もあるが（例えば文献 1), 2)），本論文で提案する推論方法に少しの変更を加えることによってこのような場合にも対応できる． □

**定義 2.2** $F$ の中に存在する負節を「一貫性制約節」（あるいは単に「制約式」）と呼ぶ．制約節は，論理式 $false \leftarrow A_1, \cdots, A_n$ で表現される．ただし，$A_k$ $(1 \leq k \leq n;\ n \geq 1)$ はアトムであり，$false$ は「矛盾」を表す． □

**例 2.1** 図 1 に示す簡単な例題プログラムを考える．$P_{ex}$ に対して，問合せ "$\leftarrow p(X,Y)$" を与える．同図の知識ベースが事実と仮説の和集合 $F \cup H$ であり，アトム $p(X,Y)$ が観測 $O$ である． □

図 2 に例 2.1 に対する SLD 反駁[15]を用いた仮説推論の実行例を示す．同図においてゴール節に現れる M なるオペレータが付いたアトム "M$A$" は，サブゴール $\leftarrow A_0$（ここでアトム $A$ は $A_0$ のある代入例）を解く際に，仮説を入力節として導出を行ったこと（すなわち，$A$ に対して仮説が立てられたこと）を示すためのマークである[5]．

ここで，図 2 の $P_{ex} \cup \{\leftarrow false\}$ の SLD 反駁木から，成功葉に現れる仮説のマルチセット $\{r(1), t(1)\}$ または $\{r(2), t(2)\}$ を仮定することは知識ベースに対して矛盾を起こすことがわかる（このような仮説のマルチセットを「矛盾仮説」と呼ぶ）．また，同図の SLD 反駁

☆ 本研究は文献 13), 14) の報告をもとにして本論文にまとめたものである．

☆☆ 本論文では，観測の説明のコストは観測の説明に現れる仮説の重みの和で与えられるものとする．

☆☆☆ 仮説推論を用いて計画問題を解く場合には，状態空間を $F$，目標を達成し得る個々の操作（部分計画）を $H$ として問題が知識表現され，$O$ として目標状態が与えられる．

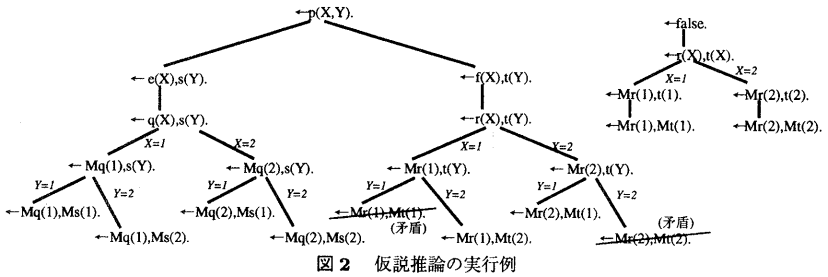



図2 仮説推論の実行例

Fig. 2 An example of abductive reasoning for $P_{ex}$.

木より，右から2番目の成功葉から得られる仮説のマルチセット $\{r(2),t(1)\}$のコストが反駁木の任意の成功葉から得られる無矛盾な仮説のマルチセットのコストにおいて最小であるので，これを説明としてゴール $\leftarrow p(X,Y)$ の最適解$\{X=2,Y=1\}$が得られる．

しかしながら，一つの最適解を得るために全解を求めるのは無駄な探索空間が大きく極めて非効率である．したがって本研究では，最適な説明の導出には無関係な探索を回避して効率的に最適解を得るために，SLD反駁に対して最適解探索アルゴリズムが持つ探索制御技術を導入する．さらに，本研究で提案する推論方法の探索制御能力を決定づけるヒューリスティック評価関数の導出方法を提案する．

なお，本論文では，反駁木に現れるゴール内のアトムの選択規則は最左優先とする．

### 2.2 関連研究

仮説推論を前節のように定義し，各仮説に好ましさの基準が与えられることによって，最適な説明を求める問題は最適解探索の問題に帰着される．これまでに提案されている最適解を保証する仮説推論システムでは，主に最良優先探索が用いられている（例えば文献9)～11) など)．

文献9)，10) では，最良優先探索の手法を用いることにより最適な説明を効率的に求める仮説推論が提案されている．しかしながら，効果的なヒューリスティック評価関数の求め方については提案されていない．

本論文では，仮説推論を実現する推論処理技術に対して A* アルゴリズム[12]の持つ探索制御技術を導入する．さらに，本仮説推論方法の探索制御能力を決定づけるヒューリスティック評価関数の導出方法を提案する．

文献16) では，実行可能条件を満たすヒューリスティック評価関数の導出方法が提案されている．しかしながら，その導出法は命題論理で表現されたプログラムを対象に説明されており，述語論理で表現されたプログラムに対する導出方法については言及されていない．プログラムが述語論理で表現されている場合には，本論文で述べるような何らかの抽象化を行わない限り導出のオーバヘッドは無視できないものと思われる．さらにその方法を，再帰的に定義された一階述語論理プログラムに対応できるように拡張するのは難しいと考える．

本論文で提案するヒューリスティック関数の導出方法は次節で提案する推論アルゴリズムを直接用いるので，システムの構成が簡単であるのみならず，任意の（再帰的に定義された）一階述語論理プログラムにも対応できる．

また，文献11) では，ビーム探索・分岐限定法および最良優先探索を用いて効率的に最適解を求める仮説推論を提案している．文献11) の研究については3.2節で述べる．

## 3. A* アルゴリズムに基づく探索制御

本節では，A* アルゴリズムの持つ探索制御技術を導入することにより効率的な推論処理方法を提案する．ここでは，簡単のため，SLD反駁を用いて本手法について説明する．

なお，本論文では，知識ベースに現れる制約式については$\leftarrow false$に対する反駁を行い，矛盾仮説のマルチセットをあらかじめ求めておくことにする．

### 3.1 A* アルゴリズムを用いたSLD反駁

本論文で提案する推論方法では，反駁木に現れるゴール節に対して以下のように定義される評価値を用いて A* アルゴリズムに基づく探索制御を実現している．

**定義 3.1** $P$ をプログラム，$O$ を与えられた観測とする．また，$P$ におけるゴール$\leftarrow O$に対する SLD 反駁木（$P\cup\{\leftarrow O\}$の SLD 反駁木と呼ぶ）における任意のゴール節 $g=\leftarrow \mathrm{M}L_1,\cdots,\mathrm{M}L_i,A_{i+1},\cdots,A_n$ について，$\hat{h}(g)$ を $g$ から成功葉に至る最適導出におけるコストの予測値，$h(g)$ を実際のコストの値とする．このとき，最適解探索における $g$ の評価値 $f(g)$ を以下

のように定義する．

$$f(g) = cost(\{L_1, \cdots, L_i\}) + \hat{h}(g)$$

$\hat{h}(g)$ は以下の条件を満足するような予測値とする．

$$0 \leq \hat{h}(g) \leq h(g)$$

$h(g) = cost(\bigcup_{j=i+1}^{n} A_j$の最適な説明) □

ここで，$cost(\{L_1, \cdots, L_i\})$ は反駁木の根から $g$ に至る導出におけるコストの値を示している．また，本論文では，$A_j$ の説明が存在しないときには，$h(g) = \infty$ とする．$\hat{h}(g)$ の値については，定義 3.4 に基づき $g$ の導出時に計算される．

**定義 3.2 A* アルゴリズムを用いた SLD 反駁**

$P$ をプログラム，$O$ を与えられた観測とする．また，$OPEN_i$, $CLOSED_i$, $SELECTED$ および $SG$ をゴールの集合とする．$OPEN_i$ および $CLOSED_i$ は，まだ展開されていないゴールの中で最適解を求めるために有効なゴールおよびもうすでに展開されたゴールの集合をそれぞれ示している．また，$SELECTED$ は現時点で最も評価値の良いゴールを要素に持つ集合である．アルゴリズムを **図 3** に示す．

ここで，ゴール $A = \leftarrow a_1, \cdots, a_n$, $B = \leftarrow b_1, \cdots, b_m$ が $A =^* B$ とは，$A$ と $B$ が variant☆ であることとし，$A \supset^* B$ とは，$A' = \leftarrow a'_1, \cdots, a'_m$（ここで，$^\forall a'_j$ $(1 \leq j \leq m) \in A'$ は $A$ に現れるアトムであるとする）と $B$ が variant であることとする． □

任意の SLD 反駁木において，すべてのゴール節 $g$ に対し条件 $\hat{h}(g) \leq h(g)$（実行可能条件と呼ぶ）が成立する時，本アルゴリズムは実行可能であり，最適なアルゴリズムである（A* の実行可能性[12]および最適性[17]より明らか）．また，本アルゴリズムは $OPEN_i$ の更新処理（図 3，9~19 行）においてループ・チェック機能を備えており，再帰構造を持つプログラムにも対応する．しかしながら，ヒューリスティック探索の持つ性質として知られているように，本アルゴリズムの性能は $g$ から成功葉へ至る導出のコストの予測値（ヒューリスティック評価関数）$\hat{h}(g)$ に大きく依存する．

したがって本論文では，実行可能条件を満たし，かつ実際のコストになるべく近いような $g$ のコストの予測値 $\hat{h}(g)$ を求める方法を次節で提案する．

### 3.2 ヒューリスティック評価関数 $\hat{h}(g)$ の導出方法

本節では，本アルゴリズムの探索制御能力を決定づけるヒューリスティック評価関数 $\hat{h}(g)$ の算出方法について述べる．

☆ 論理式 $E$,$F$ が variant とは，$E$,$F$ に対して次の単一化が成り立つことを言う（ただし $\theta$,$\sigma$ はともに最汎単一化代入 (mgu)）．$E = F\theta$ かつ $F = E\sigma$.

**A* アルゴリズムを用いた SLD 反駁**

入力: プログラム $P$, 観測 $O$
出力: 解 $Ans$: (解代入例, 最適な説明)（成功裏）または $false$（失敗裏）

```
1  begin
2    OPEN_0 := {←O}; CLOSED_0 := φ; i := 0;
3    repeat
4      SELECTED := {g_min ∈ OPEN_i | for all
                    g ∈ OPEN_i : f(g_min) ≤ f(g)};
5      各← g_min ∈ SELECTED に対して P における一段
       階の SLD 導出で得られるサブゴールをすべて求め†,
       これらの集合を SG とする;
6      CLOSED_{i+1} := CLOSED_i ∪ SELECTED;
7      SG について無矛盾性の検査††を行い P と矛盾する
       ゴールを SG から削除する;
8      OPEN' := OPEN_i \ SELECTED;
9      for each sg ∈ SG
10       if sg ⊃* ∃g ∈ OPEN' ∪ CLOSED_{i+1} then
11         if sg =* ∃g ∈ OPEN' then
12           begin
13             if f(sg) < f(g) then
14               OPEN' := OPEN' \ {g} ∪ {sg}
15           end
16         else if sg =* ∃g ∈ CLOSED_{i+1} then
17           begin
18             if f(sg) < f(g) then
19               OPEN' := OPEN' ∪ {sg}
20           end
21         else OPEN' := OPEN' ∪ {sg};
22     OPEN_{i+1} := OPEN';
23     i := i + 1;
24   until (∃g_min ∈ SELECTED は成功葉である) or
         (OPEN_i = φ);
25   if ∃g_min ∈ SELECTED は成功葉である then
         Ans := (解代入例, 最適な説明); % (成功裏に終了)
26   if OPEN = φ then Ans := false; % (失敗裏に終了)
27 end.
```

† この時点ではサブゴールの状態は未定 (unspecified) である．
†† サブゴールに現れる仮説が矛盾仮説を含むかどうかで検査する．

図 3 A* アルゴリズムを用いた SLD 反駁
Fig. 3 SLD-resolution incorporating A*.

AI における探索アルゴリズムの分野では，与えられた問題に対して実行可能なヒューリスティクスは抽象化された問題から生成することができ，さらに，このようにして得られたヒューリスティクスが効果的なものとなるには，抽象化された問題が，それを効率的に解くことができる範囲で元の問題の近似である必要があることが知られている（例えば，文献 18)）．

したがって本論文では，述語論理式で表現された知識ベース並びに問合せに対して，その近似としてそれらの述語の引数を無視した命題論理版を対象として，事前解析を行う．事前解析の精度を上げるためにはより細かいレベルまで（例えばアトムの第 1 引数まで考慮するなど）考えて解析を行えば良いが，これは事前

| 事実 | 仮説 | コスト |
|---|---|---|
| $p \leftarrow e, s.$ | $q.$ | 2 |
| $p \leftarrow f, t.$ | $r.$ | 2 |
| $e \leftarrow q.$ | $s.$ | 4 |
| $f \leftarrow r.$ | $t.$ | 2 |

図4 抽象化されたプログラム $\overline{P \backslash IC}_{ex}$
Fig. 4 Abstracted program $\overline{P \backslash IC}_{ex}$ of $P_{ex}$.

解析にかかるコストとのトレードオフの問題となる.

**定義 3.3** $P$ をプログラム，$P$ を命題論理に抽象化したものを $\bar{P}$ とする．$P$ に含まれる仮説 $E$ に対して，$\bar{E}$ のコストは以下の式で与えられる．

$$cost(\{E\}) = min(\, cost(\{L\}) \mid L \text{ は } P\text{に含まれる仮説かつ } \bar{L} = \bar{E})$$

以下，論理式（あるいはその集合）$F$ に対し，これを命題論理に抽象化したものを $\bar{F}$ で表記する． □

ここでは，抽象化の際にプログラム $P$ に対して $P$ に含まれる無矛盾性制約節（この集合を $IC$ とする）を除いたプログラム $P \backslash IC$ を考える．まず，$\overline{P \backslash IC}$ に現れるアトム $\bar{A}$ について，A* に基づく SLD 反駁（定義 3.2，図 3 参照）を実行し，$\bar{A}$ と $\bar{A}$ の最適な説明 $H_{\bar{A}}$ の組 $(\bar{A}, H_{\bar{A}})$ を求める☆．その際に用いるヒューリスティック関数は $\hat{h}(\bar{g}) = 0$ とする．次に，それらの組から得られる情報を用いてヒューリスティック評価関数 $\hat{h}(g)$ を定義する．

**例 3.1** 図 1 のプログラム $P_{ex}$ を命題論理に抽象化したプログラム $\overline{P \backslash IC}_{ex}$ を**図 4** に示す．ここで，例えば仮説 $q$ のコスト 2 は $P_{ex}$ に含まれる仮説 $q(1), q(2)$ のコストの最小値で与えられている．

$\overline{P \backslash IC}_{ex}$ に対してゴール節$\leftarrow p$,$\leftarrow e$,$\leftarrow f$,$\leftarrow q$,$\leftarrow r$,$\leftarrow s$,$\leftarrow t$をそれぞれ *OPEN* の初期集合の要素として定義 3.2 のアルゴリズムを実行する．その結果，各反駁木の成功葉よりアトムとその最適な説明の組，$(p, \{r, t\})$, $(e, \{q\})$, $(f, \{r\})$, $(q, \{q\})$, $(r, \{r\})$, $(s, \{s\})$, $(t, \{t\})$. を得る． □

**定義 3.4** $P$ をプログラム，$O$ を与えられた観測，$IC$ を $P$ に含まれる無矛盾性制約節の集合，$A$ を $P$ に含まれるアトムとし，$H_A$ を $A$ の最適な説明を表す仮説のマルチセットとする．

$P \cup \{\leftarrow O\}$の SLD 反駁木に現れるゴール節 $g = \leftarrow \mathsf{M}L_1, \cdots, \mathsf{M}L_i, A_{i+1}, \cdots, A_n$に対して，$g$ からある成功葉へ至る最適導出のコストの予測値 $\hat{h}(g)$ は以下の式で定義される．

$$\hat{h}(g) = h(\bar{g}) \qquad (*)$$

☆ アトム $\bar{A}$ の最適な説明 $H_{\bar{A}}$ が存在しない（つまり$\leftarrow \bar{A}$の反駁が存在しない）場合には，そのコストが $\infty$ となるような仮の仮説のマルチセット $H'_A$ を用いて，$(\bar{A}, H'_{\bar{A}})$ とする．

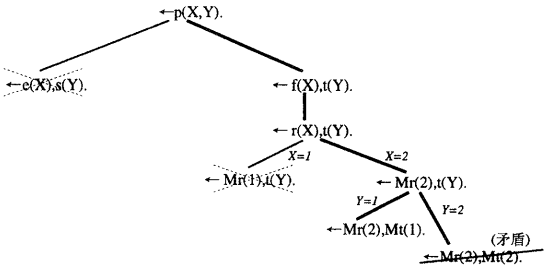



図5 $P_{ex} \cup \{\leftarrow p(X, Y)\}$の SLD 反駁木における A*による枝刈り
Fig. 5 Pruning SLD-tree for $P_{ex} \cup \{\leftarrow p(X, Y)\}$ by incorporating A*.

$h(\bar{g})$ は $\overline{P \backslash IC}$ における $\bar{g}$ に対する最適導出のコストを示す． □

定義 3.1 より式 $(*)$ は次の計算によって求められる．

$$h(\bar{g}) = cost(H_{\bar{g}}) = cost(\bigcup_{j=i+1}^{n} H_{\bar{A}_j})$$

このとき，次の性質が成立することが容易にわかる．

**定理 3.1** 式 $(*)$ から得られる予測値 $\hat{h}(g)$ は本アルゴリズムの実行可能条件 $\hat{h}(g) \leq h(g)$ を満たす．

**証明** 付録に示す． □

このようにして求められたヒューリスティック評価関数を用いることにより，A* アルゴリズムにおける実行可能条件を満足する．その結果，本アルゴリズムが最初に検出した解は必ず最適解となることが保証され，効率的な反駁を行うことが可能となる．なお，予測値 $\hat{h}(g)$ の値が $h(g)$ に近いほど，反駁時における探索空間の絞り込みが可能となるが，これは本推論システムが扱う問題に依存するものである．

**例 3.2** **図 5** に$P_{ex} \cup \{\leftarrow p(X, Y)\}$に対する A* アルゴリズムを用いた SLD 反駁の実行例を示す．同図の反駁木の成功葉から得られる仮説のマルチセット $\{r(2), t(2)\}$は問合せ $p(X, Y)$ の最適解 $p(2, 2)$ の最良な説明となる．

同図において，〰 印が付けられたゴールは，そのゴールに対する展開が枝刈りによってそれ以上行われないことを示している．また，図 2 の反駁木に比べて枝の数および M 演算子付アトムの数が少ないことから，仮説推論の過程において真と仮定される仮説の数が制限され，「合理的な仮説の選択」が実現されていることがわかる．このように，推論において生成されるゴールの数を制限することができ，仮説推論の探索空間を絞り込むことが可能となる． □

文献 11) では，ビーム探索・分岐限定法および最良優先探索の考えを用いて効率的に最適解を求める仮説推論システムが提案されている．探索を制御するパラメータとして，固定的なしきい値および動的に変化す

表1 探索空間の比較結果
Table 1 Some comparisons of search spaces.

| | 仮説生成数 | 仮説の合成回数 |
|---|---|---|
| 全解探索 | 8 | 8 |
| 本手法 | 4 | 2 |
| 文献11)の手法 | | ($\theta$=しきい値) |
| $\theta \geq 9$ | 8 | 5 |
| $\theta = 7,8$ | 8 | 4 |
| $\theta = 6$ | 8 | 3 |
| $\theta = 5$ | 7 | 2 |
| $\theta = 4$ | 6 | 1 |
| $\theta \leq 3$ | — | — |

る暫定値を用い，さらに最小コストの説明を効率的に求めるために各仮説の組合せのコストの下界値を考えている．その結果，文献11)では仮説の生成および合成の回数を制限することにより，仮説推論の探索空間を絞り込んでいる．

しかしながら，コストの下界値に関する情報が十分に活かされず，効果的なヒューリスティクスが得られない場合がある．本研究では，例2.1の知識ベース $P_{ex}$ を用いて観測 $p(X,Y)$ の最適な説明を求める仮説推論における探索空間について本手法と文献11)の推論方式との比較実験を行った．**表1**に比較結果を示す．同表から仮説の生成数についてはしきい値 $\theta$ の値に関わらず本手法の方が最適な説明に無関係な仮説の生成を回避でき，仮説の合成回数についても $\theta$ の値によっては本手法の方が無駄な合成を回避していることがわかる．文献11)の効率化手法は $\theta$ の値に大きく依存する．与えられた例題に対して何らかの解析を行わない限り $\theta$ の値を適切に求めることは困難であると考える（例2.1の問題では $\theta \leq 3$ の下では解を求めることができない）．本推論システムでは，与えられた知識ベース並びに問合せに対して命題論理のレベルで事前解析を行い，実際の仮説推論で用いられる探索制御のための情報を自動に抽出している．以上の結果より，本研究で提案されたヒューリスティック評価関数が本アルゴリズムの探索制御能力を高め，その結果，仮説推論の探索空間を効果的に絞り込んでいることがわかる．

## 4. 前向き推論を用いた実装

本節では，3節で述べた手法に基づいて，コストに基づく効率的な仮説推論を実現するための方法を述べる．本論文では，SLD反駁と本質的に等価な計算を実行する前向き推論を用いてこれを実現する．その理由は，前向き推論は，SLD反駁などの後向き推論に比べ，大量のデータを対象とする計算に適しており，大規模データベースを扱う問合せ処理システムにおいて多く用いられるからである．本研究では，例2.1の知識ベース $P_{ex}$ に含まれる仮説の数を変化させて後向き推論と前向き推論の間で推論時間の比較を行った．知識ベースの規模が大きな場合には，前向き推論の方が後向き推論に比べて5倍程度速く，有効な推論方法となることを確認している．

まず，3節で提案した推論アルゴリズムを前向き推論で実現するためにプログラム変換法を提案する．本変換法によって変換されたプログラムに対する前向き計算は，幅優先のSLD反駁をシミュレートしつつ，導出されたゴールの評価値を算出する．次に，A* アルゴリズムで行われている *OPEN*，*CLOSED*，*SELECTED* の更新手続き（定義3.2参照）を実現するプログラムを提案し，A* を用いたSLD反駁と実質的に等価な計算を実現する前向き計算手続きを定義する．

### 4.1 前向き推論のためのプログラム変換

本節では，プログラムに適切な変換を施し，その変換されたプログラムを前向き計算によって処理すると，それが元のプログラムの幅優先のSLD反駁をシミュレートしつつ，導出されたゴールの評価値を算出できることを，図1の例題知識ベース $P_{ex}$ を用いて説明する．

**例4.1** **図6**に変換されたプログラム $P_{ex}^{tr}$ を示す．ここで，$P_{ex}^{tr}$ に現れる述語 $goal(M,G,V,H,State)$ は $P_{ex} \cup \{\leftarrow p(X,Y)\}$ のSLD反駁木に現れるゴール $g$ に対応しており，述語の第1引数 $M$ は反駁木の根から $g$ に至る導出において生成された仮説のマルチセット，第2引数 $G$ は $g$ に現れるアトムのリスト，第3引数 $V$ は問合せ（ゴール）の解代入を蓄えるための変数をそれぞれ示している．また，第4引数 $H$ は $g$ の評価に関する情報を蓄えており，$cost(H)$ は $\hat{h}(g)$ の値をとる（定義3.1参照）．なお，第5引数 $State$ は $g$ の状態を表す☆ものであり，$open, closed, selected, unsp$ のいずれかの値をとる．また，$P_{ex}^{tr}$ に現れる述語 $\underline{c}(A,H_A)$ はアトム $A$ の最適な説明が $H_A$ であることを示す．

例えば，プログラム $P_{ex}^{tr}$ の上から1番目のルールは，後向き推論における確定節 $p(X,Y) \leftarrow e(X), s(Y).$ に対応しており，次の意味である．

> ゴールの状態が *selected* であるゴールの最左アトムが $p(X,Y)$ と単一化可能であり，かつ，反駁木の根（初期ゴール）からそのゴールへ至る導出のコストが $cost(M)$，そのゴールか

☆ 例えばゴール節 $g$ の状態が *open* とは，定義3.2のアルゴリズムにおいて $g$ が集合 *OPEN* の要素となることを意味する．

事実
$goal(M,[p(X,Y)|G],V,H,selected),\underline{c}(p,Hp),\underline{c}(e,He),\underline{c}(s,Hs)$
$\rightarrow goal(M,[e(X),s(Y)|G],V,H\cup He\cup Hs\setminus Hp,unsp).$
$goal(M,[p(X,Y)|G],V,H,selected),\underline{c}(p,Hp),\underline{c}(f,Hf),\underline{c}(t,Ht)$
$\rightarrow goal(M,[f(X),t(Y)|G],V,H\cup Hf\cup Ht\setminus Hp,unsp).$
$goal(M,[e(X)|G],V,H,selected),\underline{c}(e,He),\underline{c}(q,Hq)$
$\rightarrow goal(M,[q(X)|G],V,H\cup Hq\setminus He,unsp).$
$goal(M,[f(X)|G],V,H,selected),\underline{c}(f,Hf),\underline{c}(r,Hr)$
$\rightarrow goal(M,[r(X)|G],V,H\cup Hr\setminus Hf,unsp).$

仮説
$goal(M,[q(1)|G],V,H,selected),\underline{c}(q,Hq)\rightarrow goal(\{q(1)\}\cup M,G,V,H\setminus Hq,unsp).$
$goal(M,[q(2)|G],V,H,selected),\underline{c}(q,Hq)\rightarrow goal(\{q(2)\}\cup M,G,V,H\setminus Hq,unsp).$
$goal(M,[r(1)|G],V,H,selected),\underline{c}(r,Hr)\rightarrow goal(\{r(1)\}\cup M,G,V,H\setminus Hr,unsp).$
$goal(M,[r(2)|G],V,H,selected),\underline{c}(r,Hr)\rightarrow goal(\{r(2)\}\cup M,G,V,H\setminus Hr,unsp).$
$goal(M,[s(1)|G],V,H,selected),\underline{c}(s,Hs)\rightarrow goal(\{s(1)\}\cup M,G,V,H\setminus Hs,unsp).$
$goal(M,[s(2)|G],V,H,selected),\underline{c}(s,Hs)\rightarrow goal(\{s(2)\}\cup M,G,V,,H\setminus Hs,unsp).$
$goal(M,[t(1)|G],V,H,selected),\underline{c}(t,Ht)\rightarrow goal(\{t(1)\}\cup M,G,V,H\setminus Ht,unsp).$
$goal(M,[t(2)|G],V,H,selected),\underline{c}(t,Ht)\rightarrow goal(\{t(2)\}\cup M,G,V,H\setminus Ht,unsp).$
$goal(M,G,V,H,selected)\rightarrow goal(M,G,V,H,closed).$
$goal(M,G,V,H,closed)\rightarrow goal(M,G,V,H,closed).$
$goal(M,G,V,H,open)\rightarrow goal(M,G,V,H,open).$

**図6** 変換されたプログラム $P_{ex}^{tr}$
Fig. 6 Transformed program $P_{ex}^{tr}$ of $P_{ex}$.

ら成功葉へ至る最適導出のコストの予測値が $cost(H)$ であり，アトム $p,e,f$ の最適な説明がそれぞれ $Hp,He,Hf$ であるとき，ゴールの先頭のアトム $p(X,Y)$ を $e(X),f(X)$ に置き換えたサブゴールを生成する．このとき，反駁木の根からサブゴールへ至る導出のコストは $cost(M)$ であり，サブゴールから成功葉へ至る最適導出のコストの予測値は $cost(H\cup Hc\cup Hs\setminus Hp)$ となる．また，サブゴールの状態は $unsp$（未定，$unspecified$）である． □

### 4.2 A*を実現する前向き計算

次に，A*アルゴリズムが行っている $OPEN$, $CLOSED$, $SELECTED$ の更新手続き（定義 3.2 参照）を前向き計算で実現するために **図7**，**図8** に示す二つのプログラム $P_{Agg1},P_{Agg2}$ を考える．

本論文では，集合演算を行う組み込み述語 **agg**$(A,CL,S)$ を導入する．第一引数 $A$ は述語，第二引数 $CL$ は条件式から成るリスト，第三引数 $S$ は集合を示し，現在までに導出されたアトムの集合 $I$ に対して，**agg**$(A,CL,S)$ は，**(i)** $I$ から $A$ と単一化可能なアトム $A'\theta$ $(\theta=mgu(A,A'))$ をすべて求め（これらのアトムの集合を $S_r$ とする），**(ii)** $S_r$ から $CL$ に含まれる条件式をすべて満足するようなアトムをすべて取りだし，これらのアトムの集合を $S$ とする．

**定義 4.1** $P$ をプログラム，$O$ を与えられた問合せ，$IC$ を $P$ に含まれる無矛盾性制約節の集合とし，$P^{tr},P_{Agg1},P_{Agg2}$ に対する直接帰結オペレータをそれぞれ $T_{P^{tr}},T_{P_{Agg1}},T_{P_{Agg2}}$ とする．また，$\underline{C}_{\overline{P\setminus IC}}$ を $\overline{P\setminus IC}$ に含まれるアトム $A$ およびその最適な説明 $H_A$ を示す述語 $\underline{c}(A,H_A)$ からなる集合とする（例 3.1 参照）．

$P^{tr}\cup P_{Agg1}\cup P_{Agg2}\cup\underline{C}_{\overline{P\setminus IC}}$ に対して **図9** に示す手続きによって定義される前向き計算を行えば，A* に基づく探索制御を実現することができる． □

**A*を実現する前向き計算手続き**

入力: プログラム $P^{tr}\cup P_{Agg1}\cup P_{Agg2}\cup\underline{C}_{\overline{P\setminus IC}}$, 観測 $O$
出力: 解 $Ans$: (解代入例, 最適な説明) (成功裏) または $false$ (失敗裏)

```
1  begin
2    I_0:={goal({ },[O],O,H_O,selected)}
3    i := 0
4    repeat
5      I_{i+1} := T_{P_Agg2}(T_{P_Agg1}(T_{P^tr}(I_i ∪ C_{¬P\IC})));
6      I_{i+1} について無矛盾性の検査†を行い P と矛盾する
       ゴールに対応するアトム goal(M,G,V,H,State)
       を I_{i+1} から削除する;
7      i := i + 1;
8    until (goal(M,[],V,0,selected) ∈ I_i ) or
         (goal(_,_,_,_,selected) ∉ I_i);
9    if goal(M,[],V,0,selected) ∈ I_i then
         Ans := (V,M);   % (成功裏に終了)
10   if goal(_,_,_,_,selected) ∉ I_i then
         Ans := false;   % (失敗裏に終了)
11 end.
```

†$goal(M,G,V,H,State)$ の第1引数 $M$ が矛盾仮説を含むかどうかで検査する．

**図9** A*を実現する前向き計算手続き
Fig. 9 Bottom-up computation to simulate A*.

第 $i$ 回目の反復において $T_{P^{tr}}$ は 図3 のアルゴリズムの5行目の処理に対応しており，与えられたゴールから導出可能なサブゴールをすべて導出する．この時点においては，それらのサブゴールの状態は "unspecified" である．そして，与えられたゴールの状態を "closed" とし，第 $i-1$ 回目までの反復において生成されたゴールの状態が "open" または "closed" であるゴールを導出する．また，$T_{P_{Agg1}}$ は同アルゴリズムの 9-19 行目の処理に対応しており，$T_{P^{tr}}$ によって導出されたサブゴールの状態を更新す

$goal(M,G,V,H,open)$,
**agg**( $goal(M',G',\_,H',unsp), [M \supseteq M', G \supseteq^* G',\ cost(M')+cost(H') < cost(M)+cost(H)], S'), S' = \phi$
$\rightarrow goal(M,G,V,H,open)$.
$goal(M,G,V,H,unsp)$,
**agg**( $goal(M',G',\_,H',\_), [M \supset M', G \supset^* G'], S'), S' = \phi$,
**agg**( $goal(M'',G'',\_,H'',open), [M = M'', G =^* G'',\ cost(M'')+cost(H'') \leq cost(C)+cost(H)], S''), S'' = \phi$,
**agg**( $goal(M''',G''',\_,H''',closed), [M = M''', G =^* G''',\ cost(M''')+cost(H''') \leq cost(C)+cost(H)], S'''), S''' = \phi$
$\rightarrow goal(M,G,V,H,open)$.
$goal(M,G,V,H,closed)$ $\rightarrow goal(M,G,V,H,closed)$.

**図7** プログラム $P_{Agg1}$
Fig. 7 Program $P_{Agg1}$.

$goal(M,G,V,H,open)$,
**agg**( $goal(Mo,\_,\_,Ho,open), [cost(Mo)+cost(Ho) < cost(C)+cost(H)], S), S = \phi$
$\rightarrow goal(M,G,V,H,selected)$.
$goal(M,G,V,H,open), goal(Mo,\_,\_,Ho,open), cost(Mo)+cost(Ho) < cost(C)+cost(H)$
$\rightarrow goal(M,G,V,H,open)$.
$goal(M,G,V,H,closed)$ $\rightarrow goal(M,G,V,H,closed)$.

**図8** プログラム $P_{Agg2}$
Fig. 8 Program $P_{Agg2}$.

る．$T_{P_{Agg2}}$ は同アルゴリズムの4行目の処理に対応しており，$^{\forall}goal(M,G,V,H,open) \in I_i$ のなかで評価値 $cost(M)+cost(H)$ が最小であるアトムを選びそのアトムの状態を $selected$ に移す．

例えば，$P_{ex}^{tr} \cup P_{Agg1} \cup P_{Agg2} \cup C_{\overline{P \setminus IC}_{ex}}$ に対してゴールに相当するアトム $goal(\{\ \}, [P(X,Y)], P(X,Y), \{r,t\}, selected)$ を与えて，定義 4.1 の前向き計算を行えば，図5と同様にして，無駄な探索を絞り込むことができる．

## 5. 実験結果

本論文で提案した効率化法の有効性を確認するために，推論時間の比較実験を行った．例題としては，ロボットの荷物運搬作業に関する計画問題を用いた．仮説は，ロボットの一行程の作業とし，仮説のコストは，その一行程にかかる作業時間で与えた．また，知識ベースに現れる仮説の数を k とし，k をパラメータとして知識ベースの規模を変化させて実験した．

実験結果を **図10** に示す．実験は計算機 SUN4/75 上で言語 SICStus Prolog を用いて行った．実線は本論文で提案した手法，破線はヒューリスティック評価関数を用いない（すなわち $\hat{h}(g)=0$）最良優先探索に基づく手法の結果を示している．知識ベースの規模が増大するほど，本手法を用いることにより，推論時間がそれぞれ大幅に短縮されることがわかる．また，ヒューリスティック評価関数 $\hat{h}$ を求めるための事前解析自身に要する時間は，命題論理のレベルに抽象化してプログラム解析を行っているため，知識ベースの規模 k が大きい場合には，実際の仮説推論に要する時間の約1%以下であり，無視できる程度であった．

## 6. おわりに

コストに基づく仮説推論において効率的な最適解探索法を提案し，一階述語論理ホーン節を対象とする仮説推論システムを実現した．

本論文で提案した仮説推論システムは，論理プログラミングの分野における問合せ処理技術に対して A* アルゴリズムの持つ探索制御技術を導入することにより，与えられた観測に対して最良の説明を効率的に求めるものである．本論文で導入した A* アルゴリズムを代表とするヒューリスティックな探索法においては，その探索制御能力はヒューリスティック評価関数 $\hat{h}$ に大きく依存する．そこで本論文では，実行可能条件を満たし，かつ，実際のコスト $h$ になるべく近い値をとるような，ヒューリスティック評価関数を求めるために，プログラム解析を用いた $\hat{h}$ の導出方法を提案した．さらに，本論文で提案した仮説推論システムを前向き推論を用いて実装し，その有効性を確認した．本推論システムの実装法は，前向き計算の一制御法として捉えることができ（例えば文献19)），演繹データベースの分野においても興味深いものであると思われる．

本論文では，観測の説明は仮説マルチセットで表現されるとしたが，定義 3.1 を少し変更することにより，説明が通常の仮説集合で表現される場合にも対応できる．

また，本論文では，仮説の選択の基準として各仮説に与える重みは正の数とし，観測の説明のコストが観測の説明に現れる仮説の重みの和で与えられる場合について考えた．しかしながら，仮説推論の応用例題として知られる診断問題等では，各仮説の重みは確率で表されることが多い（例えば，文献8)～10) など）．本

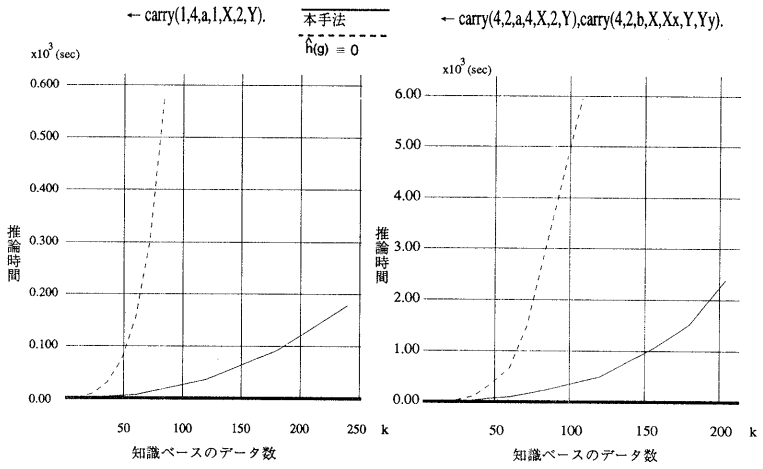



**図 10** 実験結果
Fig. 10 Experimental results.

論文で提案した仮説推論システムは，最適解探索として一般性の高い A* アルゴリズムの考えを導入しているため，定義 3.2 のアルゴリズムを少し変更することにより，確率に基づく推論における最適解探索にも対応できる[14]．

A* アルゴリズムは，ヒューリスティックな評価関数を最も効率的に利用できる最適解探索法として広く知られているが，探索の制御に用いる集合 $OPEN, CLOSED$ の保持のために必要な記憶量が極めて大きいという問題点を持つ．したがって，今後の課題としては，A* の改良アルゴリズムである IDA*[20] を本手法に適用することが挙げられる．

**謝辞** この研究は，ICOT 並列アブダクション WG（主査・古川康一 慶応義塾大学教授）における討論を契機に行ったものである．石塚 満 東京大学教授をはじめとする同 WG の委員諸氏に深謝いたします．また，本研究で提案した事前解析法の実行可能性について数多くのご助言をいただいた，原尾政輝 九州工業大学教授，馬場口登 大阪大学助教授に深く感謝いたします．なお，本研究は，一部，(財) 電気通信普及財団の助成により行われた．

## 参 考 文 献

1) Poole, D.: A Logical Framework for Default Reasoning, *Artif. Intell.*, Vol. 36, pp. 27–47 (1988).
2) 井上克巳：アブダクションの原理，人工知能学会誌，Vol. 7, No. 1, pp. 48–59 (1992).
3) Selman, B. and Levesque, H.J.: Abductive and Default Reasoning: A Computational Core, *Proc. of the AAAI-90*, pp. 343–348 (1990).
4) 石塚 満：仮説推論の計算量と高速化メカニズム，人工知能学会誌，Vol. 9, No. 3, pp. 342–349 (1994).
5) 加藤昇平，世木博久，伊藤英則：プログラム解析に基づく仮説推論の高速化技法，情報処理学会論文誌，Vol. 35, No. 10, pp. 2019–2028 (1994).
6) Kato, S., Seki, H. and Itoh, H.: An Efficient Abductive Reasoning System Based on Program Analysis, Cousot, P. et al. eds., *Static Analysis, Proc. of the 3rd Intl. Workshop*, Lecture Notes in Computer Science, Vol. 724, pp. 230–241, Springer-Verlag, Padova (1993).
7) Ohta, Y. and Inoue, K.: A Forward-Chaining Hypothetical Reasoner Based on Upside-Down Meta-Interpretation, *Proc. of the Intl. Conf. on FGCS '92*, pp. 522–529 (1992).
8) Poole, D.: Probabilistic Horn abduction and Bayesian networks, *Artif. Intell.*, Vol. 64, pp. 81–129 (1993).
9) Charniak, E. and Shimony, S. E.: Probabilistic semantics for cost based abduction, *Proc. of the AAAI-90*, pp. 106–111 (1990).
10) Poole, D.: Logic Programming, Abduction and Probability, *Proc. of the Intl. Conf. on FGCS '92*, pp. 530–538 (1992).
11) 近藤朗子，石塚 満：述語論理知識を扱う仮説推論における最適解の高速推論法，人工知能学会誌，Vol. 9, No. 2, pp. 110–118 (1994).
12) Nilsson, N. J.: *Principles of Artificial Intelligence*, Tioga, Palo Alto, CA (1980).
13) Kato, S., Seki, H. and Itoh, H.: Cost-based Horn Abduction and its Optimal Search, *Proc. of the 3rd Intl. Conf. on Automation, Robotics and Computer Vision*, Singapore, pp. 831–835 (1994).

14) Kato, S., Seki, H. and Itoh, H.: Cost-based Horn Abduction to Focus on the Most Probable Diagnosis, *Proc. of the 5th Intl. Workshop on Principles of Diagnosis*, New Paltz, NY, pp. 148–152 (1994).
15) Lloyd, J. W.: *Foundations of Logic Programming*, Springer (1984). Second, extended edition (1987).
16) Charniak, E. and Husain, S.: A New Admissible Heuristic for Minimal-Cost Proofs, *Proc. of the AAAI-91*, pp. 446–451 (1991).
17) Dechter, R. and Pearl, J.: Generalized Best-First Search Strategies and the Optimality of A*, *J. Assoc. Comput. Math.*, Vol. 32, No. 3, pp. 505–536 (1985).
18) Prieditis, A. E.: Machine Discovery of Effective Admissible Heuristics, *Machine Learning*, Vol. 12, pp. 117–141 (1993).
19) Ramakrishnan, R., Srivastava, D. and Sudarshan, S.: Controlling the Search in Bottom-up Evaluation, *Joint Intl. Conf. and Symposium on Logic Programming*, pp. 273–287 (1992).
20) Korf, R. E.: Depth-First Iterative-Deepening: An Optimal Admissible Tree Search, *Artif. Intell.*, Vol. 27, pp. 97–109 (1985).

## 付録 定理 3.1 の証明

$P$ をプログラム，$O$ を与えられた観測，$IC$ を $P$ に含まれる無矛盾性制約節の集合とし，$A$ を $P$ に含まれるアトムとする．また，$\hat{h}(g)$ を$P \cup \{\leftarrow O\}$の SLD 反駁木に現れるゴール節 $g = \leftarrow \mathsf{M}L_1, \cdots, \mathsf{M}L_i, A_{i+1}, \cdots, A_n$からある成功葉へ至る最適導出のコストの予測値とする．

定理 3.1 を証明するために，まず次の二つの補題を示す．

**補題 3.1** $H$ を仮説マルチセットとする．このとき，以下の式が成立する．

$^{\forall}h \in H$ に対して$^{\forall}\bar{h} \in \bar{H}$ かつ $cost(\{h\}) \geq cost(\{\bar{h}\})$

**証明** 定義 3.3 より明らか． □

**補題 3.2** $P$ をプログラム，$IC$ を $P$ に含まれる無矛盾性制約節の集合，$\leftarrow G$をゴールとし，$\Delta$ を M 演算子付アトムの連言とする．ここで以下の条件が成り立つと仮定する．

- $P \cup \{\leftarrow G\}$の SLD 反駁木において，ゴール $\leftarrow \Delta$を導出するような$\leftarrow G$の反駁が存在する．

このとき，$\overline{P \backslash IC} \cup \{\leftarrow \bar{G}\}$の SLD 反駁において，ゴール$\leftarrow \bar{\Delta}$を導出するような$\leftarrow \bar{G}$の反駁が必ず存在する．

**証明** SLD 反駁の長さに関する帰納法により明らか． □

**定理 3.1** 式 (*) から得られる予測値 $\hat{h}(g)$ は実行可能条件 $\hat{h}(g) \leq h(g)$ を満たす．

$$\hat{h}(g) = h(\bar{g}) \qquad (*)$$

**証明** $H_O$ を $P$ における $O$ の最適な説明を示す仮説のマルチセット，$H_{\bar{O}}$ を $\overline{P \backslash IC}$ における$\bar{O}$ の最適な説明を示す仮説のマルチセットとする．補題 3.2 より $\overline{H_O}$ は $\bar{O}$ の説明となるので以下の不等式が成立する．

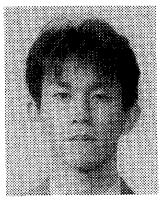
$$cost(\overline{H_O}) \geq cost(H_{\bar{O}}) \qquad (1)$$

一方で，補題 3.1 より以下の不等式が成立する．

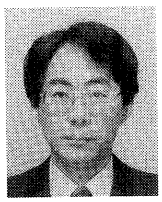
$$cost(H_O) \geq cost(\overline{H_O}) \qquad (2)$$

(1),(2) より $cost(H_O) \geq cost(H_{\bar{O}})$ が成立し，定義 3.1 より $h(\leftarrow O) \geq h(\leftarrow \bar{O})$ が成り立つ．さらに式 (*) より $h(\leftarrow O) \geq \hat{h}(\leftarrow O)$ が成立する．ここで，ゴール $\leftarrow O$ を $g = \leftarrow \mathsf{M}L_1, \cdots, \mathsf{M}L_i, A_{i+1}, \cdots, A_n$ で置き換えれば定理 3.1 は明らかに成立する． □

(平成 7 年 1 月 27 日受付)
(平成 7 年 7 月 7 日採録)



**加藤 昇平**（学生会員）
1970 年生．1993 年名古屋工業大学電気情報工学科卒業．1995 年同大学院工学研究科電気情報工学専攻博士前期課程修了．現在同大学院工学研究科博士後期課程在学中．高次推論，論理プログラミング等に興味を持つ．人工知能学会会員．



**世木 博久**（正会員）
1979 年東京大学工学部計数工学科卒業．1981 年同大学院工学系研究科修士課程修了．同年 4 月より三菱電機 (株) 中央研究所に勤務．1985 年～1989 年（財）新世代コンピュータ技術開発機構に出向．1992 年 4 月より名古屋工業大学知能情報システム学科助教授．工学博士．論理プログラミング，演繹データベース等に興味を持つ．電子情報通信学会，人工知能学会，ACM，IEEE Computer Society 各会員．

**伊藤　英則**（正会員）

1974 年名古屋大学大学院工学研究科博士課程電気・電子専攻満了．工学博士号取得．同年日本電信電話公社入社，横須賀研究所勤務．1985 年（財）新世代コンピュータ技術開発機構出向．1989 年より名古屋工業大学教授，現在知能情報システム学科所属．これまでに，数理言語理論とオートマトン，計算機ネットワーク通信 OS，知識ベースシステムなどの研究と開発に従事．電子情報通信学会，人工知能学会，ファジー学会各会員．