プログラム解析に基づく仮説推論の高速化技法

加藤昇平[†]　世木博久[†]　伊藤英則[†]

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，AI の分野において多くの研究が報告されている．しかるに，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．仮説推論においては，与えられた観測（ゴール）を説明する仮説集合が知識ベース内の無矛盾性制約と矛盾しないかという無矛盾性の検査処理が必要となる．この無矛盾性検査処理は，通常与えられた観測（ゴール）とは無関係に，知識ベースに対して大域的に行われるので，その結果仮説推論の計算量を増大させる要因となっている．従って本論文では，一階述語論理を用いた仮説推論の高速化を実現するために，プログラム解析に基づく無矛盾性の検査の効率化を提案する．本論文で提案するプログラム解析法は，ホーン節述語論理で表現された知識ベースを命題論理のレベルに抽象化することにより，実際の推論において計算すべき探索空間を近似的に求めるものである．また，プログラム解析結果を利用して効率的な仮説推論システムを実現するためにマジックセット法を拡張する．本論文で述べる効率化法をマジックセット法に基づく前向き推論を用いて計算機上に実装し，その実験結果についても報告する．

An Efficient Abductive Reasoning System Based on Program Analysis

Shohei Kato,[†] Hirohisa Seki[†] and Hidenori Itoh[†]

Abductive reasoning has attracted much attention in the field of AI and many interesting applications such as diagnosis, scheduling and design. It is, however, known to be computationally very expensive for large problems. In abductive reasoning, consistency conditions a derived solution should satisfy are usually given in the form of negative clauses (i.e., headless clauses). Checking the consistency of solutions wrt negative clauses then becomes a global one, thus losing the goal-directedness. In this paper, we propose a framework of static analysis for first-order abductive reasoning with consistency checking, to analyze the logical dependency between a goal and each relevant negative clause approximately, thereby pruning unnecessary search involving the consistency checking. We then explain the implementation of our abductive reasoning system, based on the magic sets transformation. The original magic sets method is extended to do abductive reasoning efficiently, taking into account the results obtained from the program analysis. We also show some empirical results.

1. はじめに

仮説推論の研究に関しては，その理論的枠組の提案，効率的な推論システムの実現方法，ならびに診断・設計問題等への応用など，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]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，$F\cup h$ から O が説明できて（すなわち，$F\cup h\vdash O$），かつ $F\cup h$ は無矛盾である（すなわち，$F\cup h\nvdash\square$）ような仮説の集合 h 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と，論理プログラミングや演繹データベースの分野においてこれまでに研究されてきた問合せ処理における様々な最適化技術が適用できることは，容易に予想される．実際に，文献 4)，5)，6) などの研究においては，演繹データベースの分野で提案された上昇型計算法（例えばマジックセット法[7]）や下降型計算法（例えば QSQ 法[8] など）を仮説推論に適用し，それらの問合せ処理が持つ，（i）同一のサブゴールを繰り返し解く重複計算を排除し，（ii）与えられたゴールに無関係な推論を

† 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

回避するという「ゴール指向」(goal-directed) の推論を行う性質を利用して，効率の良い仮説推論を提案している．

しかるに，仮説推論と演繹データベースにおける通常の問合せ処理とが異なる点としては，与えられた観測を説明する仮説集合が，知識ベース内の無矛盾性制約と矛盾しないかという無矛盾性の検査を受けなければならないことである．この無矛盾性検査のための推論は与えられたゴールとは無関係に大域的に行われるので，マジックセット法や QSQ 法の持つ「ゴール指向性」の利点が失われることになる．

従って本論文では，この問題を解決するために，知識ベース（プログラム）に出現する述語の依存関係を解析することによる仮説推論の効率化技法を提案する[☆]．本効率化技法の特徴は，与えられたゴールに無関係な無矛盾性制約を検出するのみならず，ゴールに関係がある無矛盾性制約に対しても，それに対する無矛盾性検査時の不必要な探索空間の刈り込みを試みる点にある．本論文で述べる効率化法を計算機上に実装し，その実験結果についても報告する．

まず2章で従来の仮説推論の方式について述べ，その問題点を指摘し，3章でその問題点を解決するプログラム解析法を後向き推論の枠組で説明し，4章でこの解析法のマジックセット法に基づく前向き推論による実装化について述べる．そして，5章で実験結果について説明する．

2. 仮説推論

2.1 仮説推論の定式化

ここでは，本論文で用いる Poole により提案された仮説推論の枠組[1]について説明する．

定義 2.1[1] ホーン節集合 F（「事実」と呼ぶ）と，単位節の集合 H（「仮説集合」と呼ぶ）が与えられたとする．また，存在束縛されたアトムの連言 O（「観測」，または単に「問合せ」と呼ぶ）が与えられたとする．このとき，O の $F \cup H$ による説明とは，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集合 h を求めることである[☆☆]．

$F \cup h \vdash O$（$F \cup h$ から O が証明される）（AR 1）

$F \cup h \nvdash \square$（$F \cup h$ は無矛盾である）　　（AR 2）□

ここで，F は確定節と負節からなる集合であり常に成り立つ知識として扱われる．一方，H の要素の代入例からなる集合は F と矛盾する可能性がある．

定義 2.2 F の中に存在する確定節を「ルール」と呼び，F の中に存在する負節を「無矛盾性制約節」（あるいは単に「制約節」）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．また，制約節が F 内に複数個ある場合は，$false_i$ $(i \geq 1)$ のように添字をつけて互いに区別する．

例 2.1 図1に示す簡単な例題[11]を考える[☆☆☆]．部門 s1 の一人Xと部門 S2 の一人Yが会議室Zで会議を行う（述語 m(X, Y, Z) で表す），あるいは，ラウンジZで談合を行う (d(X, Y, Z) で表す) 場合のスケジューリング問題である．

ルール：

-会議を行うには部門 s1 のXと s2 のYが出席すると仮定でき（それぞれ hp(X, s1) と hp(Y, s2) で表す），かつ会議室Zが空いている（v(Z) で表す）ことが条件であり，談合を行うには s1 のXと s2 のYが出席すると仮定でき，かつラウンジが静かである（q(Z) で表す）ことが条件である．図1では (1)，(3)，(2)，(4) のルールで表現されている（以下同様）．

-会議室は 101，102 の2室，ラウンジは 201 から 204 の4室が用意されている．　(DB)

-会議 101 およびラウンジ 204 は利用できないことがわかっている．　(DB)

ルール				
m(X,Y,Z) ←	hp(X,s1),hp(Y,s2),v(Z).			(1)
d(X,Y,Z) ←	hp(X,s1),hp(Y,s2),q(Z).			(2)
v(Z) ←	r(Z),hv(Z).			(3)
q(Z) ←	l(Z),hq(Z).			(4)
a(Z) ←	r(Z),hv(Z).			(5)
a(Z) ←	l(Z),hq(Z).			(6)
r(101).	r(102).			(DB)
l(201).	l(202).	l(203).	l(204).	(DB)
na(101).	na(204).			(DB)
仮説集合				
hv(Z).	hq(Z).			(H)
hp(b,s1).	hp(c,s1).	hp(e,s2).	hp(f,s2).	(H)
無矛盾性制約				
$false_7$ ←	hp(X,D),nhp(X,D).			(7)
$false_8$ ←	a(Z),na(Z).			(8)

図 1 例題知識ベース P_{ex}

Fig. 1 An example : P_{ex}.

☆ 本稿は文献 9)，10) の報告をもとにして本論文にまとめたものである．

☆☆ 一般に，条件 (AR 1)，(AR 2) を満たすような h は複数存在するが，本論文では，このような h をすべて求める全解探索について考える．

☆☆☆ この例は非再帰的なルールで表現されているが，本論文で提案する仮説推論システムは再帰的な知識ベースにも対応できる．再帰的な知識ベースに対する適用例としては文献 10) を参照されたい．

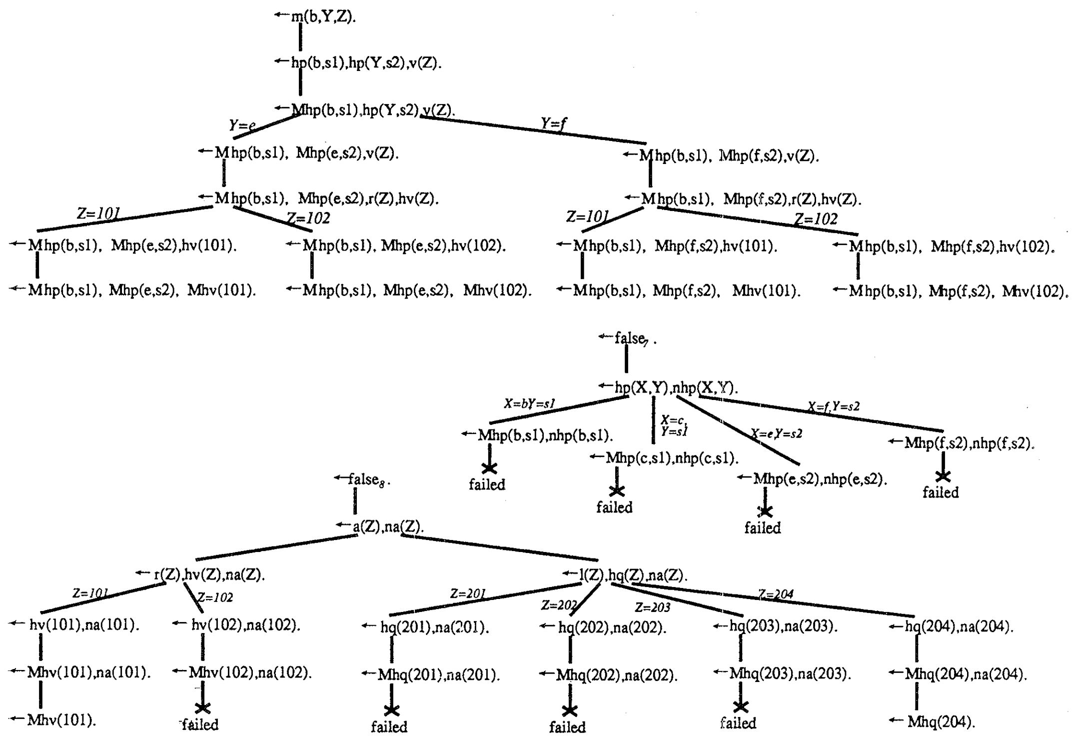


図 2　仮説推論の実行例
Fig. 2 An example of abductive reasoning for P_{ex}.

仮説集合：

-通常会議室は空いており，ラウンジは静かである．また，部門 s1 の b は通常，会議および談合に出席でき，これを hp(b, s1) で表す（c, e, f についても同様である）．　(H)

無矛盾性制約：

-部門DのXが，会議もしくは談合に出席できるであろうと仮説が立てられているのに出席できない時，知識ベースは矛盾する．　(7)

-「利用できる」と仮定された会議室またはラウンジが利用できない時，知識ベースは矛盾する．　(8)，(5)，(6)

ここで，「部門 s1 の b は誰とどこで会議を行うことができるか」というゴール "$\leftarrow m(b, Y, Z)$" が与えられたとする．なお，単位節以外の節を識別するために，識別番号として各節に自然数を割り当てる．この番号を「節番号」と呼ぶ．図1においては（ ）内の数字が，対応する節の節番号であるとする．

図1の知識ベースが事実と仮説の和集合 $F \cup H$ であり，ゴール "$\leftarrow m(b, Y, Z)$" が観測 O である．　□

図 2 に例 2.1 に対する OLDT 反駁[12]を用いた仮説推論の実行例を示す．OLDT 反駁とは，QSQ 法等と同様に通常の SLD 反駁にループチェック機能とレンマの利用による解の再利用の機能を導入したものである☆．また，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，論理的には *true* を表し，記号 **M** は単に，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）を記録するためのマークである．

図2において，ゴール$\leftarrow m(b, Y, Z)$ に対する反駁木の一番左側の枝から，仮説集合 $\{hp(b, s1),\ hp(e, s2),\ hv(101)\}$ の説明の下で解 $m(b, e, 101)$ が得られる．しかしながら，ゴール$\leftarrow false_8$ に対する反駁木から，仮説 $hv(101)$ を立てる，または，仮説 $hq(204)$ を立てることは知識ベースに対し矛盾することがわかる．従って，解 $m(b, e, 101)$ に対する説明は矛盾するのでこの解は却下される．

一方で，ゴール$\leftarrow m(b, Y, Z)$ に対する反駁木の左から二番目の枝からは，仮説集合 $\{hp(b, s1),\ hp(e, s2),$

☆ ここでは説明の簡単のため，解テーブルは省略する．

$hv(102)\}$ が無矛盾であるので，この仮説集合を説明として解 $m(b,e,102)$ が得られる．

このように仮説推論では，与えられた観測を説明する仮説集合は，知識ベース内の無矛盾性制約と矛盾しないかという無矛盾性の検査を受けなければならない．しかし，この無矛盾性の検査のための推論は，必ずしも効率的ではなく，時には無駄な処理を伴う．

2.2 関連研究とその問題点

これまでに提案されている仮説推論システムとしては，下降型（後向き）推論に基づくもの[6),13)]や，マジックセット法などの上昇型（前向き）計算を利用したもの[4),5)]などがある．これらのシステムでは，観測に対する推論において真であると仮定された仮説の集合に対する無矛盾性の検査は，制約節（負節）に対する反駁木を形成して矛盾を引き起こす仮説集合を求め，これらの集合と仮定された仮説の集合との包含関係を調べることによって行われる☆．その結果，制約節の評価時には，与えられたゴールとは無関係な部分の反駁木をも計算してしまう可能性がある．

例えば，図2においてゴール$\leftarrow false_8$ に対する反駁木の右側の部分木より，仮説 $hq(204)$ を立てることは知識ベースに対し矛盾することがわかるが，ゴール$\leftarrow m(b,Y,Z)$ に対する反駁木において，仮説 $hq(204)$ は全く現れない．従って，この部分木はゴール$\leftarrow m(b,Y,Z)$ には無関係であり，このような部分木に対し推論を行うことは無駄である．同様にして，ゴール$\leftarrow false_7$ については，もはや反駁を行う必要がないことがわかる．すなわち，仮説推論においては無矛盾性の検査が要求されるために，問合せ処理における「ゴール指向性」を失っていることになる．

このような問題点に関しては，文献 5) を除き，ほとんど研究がなされていない．文献 5) では，与えられたゴールに無関係な制約節を検出する方法が提案されている．例えば図2において，ゴール$\leftarrow false_7$ に対する反駁を回避することができるが，ゴール$\leftarrow false_8$ に対しては全計算をしてしまい，仮説推論時において「ゴール指向性」を失っている．

☆ 無矛盾性の検査については，問合せ処理の後に，観測に対する推論において仮定された仮説の集合と事実を用いて前向き推論を行い，矛盾 (false) が導出されるかどうかを調べる方法があるが（例えば，文献13)），本論文では，後向き推論を用いて問合せ処理と同時に制約節を評価する方法を取ることにより，問合せ処理における反駁木の形成途中でも，あるゴールに矛盾が生じた場合はこれを早めに（成功葉に至る以前に）刈り込むことを可能にしている．

そこで本論文では，上記の問題点を解決する一方法について以下の章で説明する．

3. プログラム解析を用いた探索空間の枝刈り

本論文では，無矛盾性制約節の評価時における「ゴール指向性」を実現するためのプログラム解析法を提案する．

ここでは，説明のため OLDT 反駁を用いて本解析法を提案し，次章において，マジックセット法に基づいた前向き推論を用いた本解析の実装について述べる．

本論文で提案するプログラム解析法は，以下の2つの処理からなる．

(1) **仮説依存関係の解析** 与えられた観測に対する説明がどのような仮説に依存するかを近似的に解析する．同様に，知識ベース内の無矛盾性制約節についても，それがどのような仮説に依存するかを近似的に解析する．

(2) **仮説依存関係による制約節の検査** 仮説依存関係の解析結果から，無矛盾性制約節の評価時における無駄な探索を検出する．

与えられた問合せと無矛盾性制約節との論理的な依存関係を近似的に解析するために，本論文では述語論理式で表現された知識ベースならびに問合せに対して，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，これは事前解析にかかるコストとのトレードオフの問題となる．

例 3.1 例 2.1の知識ベース P_{ex}（図1参照）を命題論理へ抽象化した知識ベース $\bar{P}_{ex}$ を，**図3**に示す． □

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

ルール

1:	$m \leftarrow hp, hp, v.$
2:	$d \leftarrow hp, hp, q.$
3:	$v \leftarrow r, hv.$
4:	$q \leftarrow l, hq.$
5:	$a \leftarrow r, hv.$
6:	$a \leftarrow l, hq.$
	$r.$
	$na.$
	$l.$

仮説集合

$hp.$
$hv.$
$hq.$

無矛盾性制約

7:	$false_7 \leftarrow hp, nhp.$
8:	$false_8 \leftarrow a, na.$

図 3 命題論理に抽象化された知識ベース $\bar{P}_{ex}$
Fig. 3 Abstracted program $\bar{P}_{ex}$ of P_{ex}.

3.1 仮説依存関係の解析

本論文での解析を行うために，OLDT 反駁に以下のような変更を加える．

入力節情報付 OLDT 反駁

- ゴール G の反駁とは，G の反駁木における根から，空節もしくは $\mathbf{M}$ 演算子付アトムのみから成るゴール節$\leftarrow \mathbf{M}h_1, \cdots, \mathbf{M}h_k$ ($k \geq 1$, h_i はアトム) でラベル付けられたノード（成功葉と呼ぶ）までの経路であり，成功葉に現れる仮説 h_i の集合 $\{h_1, \cdots, h_k\}$ を G の「候補仮説」，この経路において選択された入力節番号の集合を G の「入力節集合」と呼ぶ．
- 解テーブルはエントリの集合で，各エントリは，キーと解リストの組からなる．キーはアトムである．解リストは，対応するキーの候補仮説と，入力節集合の組から成るリストである．

図 4 に抽象化された知識ベース $\overline{P}_{ex}$ におけるゴール$\leftarrow m$ および$\leftarrow false_i$ ($i=7, 8$) に対する OLDT 反駁木を示す．

3.2 仮説依存関係による制約節の検査

事実 F および仮説集合 H の和集合であるプログラム P とゴール$\leftarrow O$ に対して，それを命題論理に抽象化したものをそれぞれ $\overline{P}$, $\leftarrow\overline{O}$ とする．また，P におけるゴール$\leftarrow O$ に対する OLDT 反駁木 ($P \cup \{\leftarrow O\}$ の OLDT 反駁木と呼ぶ) において，O のある候補仮説を H_O とする (つまり $F \cup H_O \vdash O$ である)．同様にゴール$\leftarrow false$ に対しても，P における $false$ のある候補仮説を H_{false} とする．この時，組 (H_O, H_{false}) で，H_{false} が H_O の部分集合となるようなものを，O に対する「矛盾仮説対」と呼ぶことにする ($(\overline{H}_{\overline{O}}, \overline{H}_{false})$ に対しても同様に定義する)．このように定義すると，次の命題が成り立つことが分かる（証明は付録に示す）．

命題 3.1 P をプログラム，P に含まれる事実を F，仮説集合を H とし，O を観測とする．O に対する候補仮説 H_O が条件 (AR 2) を満たしていないとする．このとき，(ⅰ) $P \cup \{\leftarrow false\}$ の反駁木において，組 (H_O, H_{false}) が O に対する矛盾仮説対となるような H_{false} を $false$ の候補仮説として持つ成功葉が必ず存在し，(ⅱ) (ⅰ) が成り立つならば，$(\overline{H}_{\overline{O}}, \overline{H}_{false})$ も $\overline{O}$ に対する矛盾仮説対となる．□

言い換えると，$P \cup \{\leftarrow O\}$ の仮説推論では，候補仮説 H_O に対する無矛盾性の検査においては，まず命題論理版において，すべての $\overline{H}_{\overline{O}}$ に対して $(\overline{H}_{\overline{O}}, \overline{H}_{false})$ が矛盾仮説対であるか否かを調べ，もしそれが矛盾仮説対でないことが事前にわかるならば，このような候補仮説 H_{false} を仮定する導出は行う必要がないことになる．

例 3.2 $\overline{P}_{ex} \cup \{\leftarrow false_8\}$ の OLDT 反駁木の右側の枝から候補仮説 {hq} を得る（図 4 参照）．また，$\overline{P}_{ex} \cup \{\leftarrow m\}$ の OLDT 反駁木から得られる$\leftarrow m$ の候補仮説は {hp, hv} である．従って，組合せ（{hp, hv}, {hq}）は矛盾仮説対ではないので，この枝は，問合せ“$\leftarrow m$”に関しては無関係な枝である．

また $\overline{P}_{ex} \cup \{\leftarrow false_7\}$ の OLDT 反駁木からは，実際の仮説推論において，ゴール$\leftarrow false_7$ の OLDT 反駁は行う必要がないことがわかる．これは，文献 5) の効率化に対応しており，我々の方法は，文献 5) を特別な場合として含んでいる．□

この例からわかるように，命題論理に抽象化した知識ベースに対する OLDT 反駁木から，問合せに無関係な探索を検出することができる．述語論理版の実際の仮説推論において不必要な探索を回避するため，候補仮説に付随する入力節集合を用いる．

例 3.3 図 4 において $\overline{P}_{ex} \cup \{\leftarrow false_8\}$ の OLDT 反駁木の左側の成功葉から得られる入力節集合は {8, 5} である．実際の $P_{ex} \cup \{\leftarrow false_8\}$ の OLDT 反駁において，使用する入力節をその節番号が {8, 5} で与えられる節に限定すると，**図 5** に示すように探索空間を絞り込むことができる．□

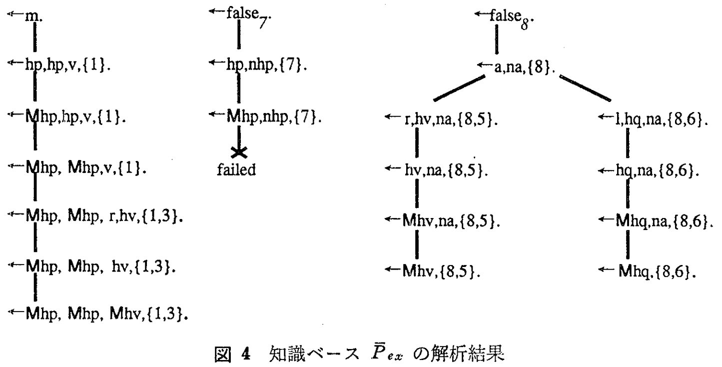


図 4 知識ベース $\overline{P}_{ex}$ の解析結果
Fig. 4 Abstracted *OLDT*-trees corresponding to those in Fig. 2.

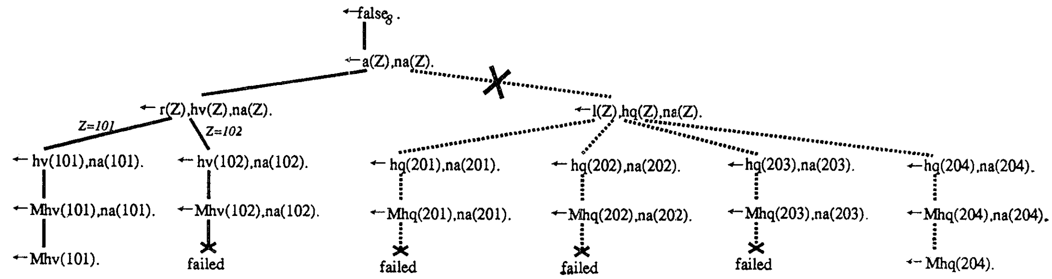


図 5 制約節の評価時における枝刈りの例
Fig. 5 Pruning OLDT-tree for $P_{ex} \cup \{\leftarrow false_8\}$.

4. マジックセット法による枝刈りの実現

この章では，3章で述べた仮説依存関係の解析と，その解析に基づいた仮説推論を実現するための方法を述べる．本論文では，OLDT 反駁と本質的に等価な計算を実行するマジックセット法[7]に基づいた前向き推論を用いてこれらを実現する*．その理由は，マジックセット法に基づいた前向き推論は，OLDT 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（大規模な知識ベースを対象とした問合せに対する全解探索など）に適しているからである．

通常のマジックセット法は，与えられたホーン節プログラムおよび問合せにプログラム変換を適用し，その変換されたプログラムに対する前向き推論が与えられたプログラムに対する後向き推論をシミュレートできるようなプログラムを導出する．変換されたプログラムおよび問合せに対して不動点を求める前向き推論は，問合せに対して全解探索を行う後向き推論に相当する．また，前向き推論の終了条件を変更する**ことにより，問合せに対する単解探索にも対応できる．

4.1 マジックセット法を用いた仮説依存関係の解析法

まず，OLDT 反駁によるプログラム解析法は，その解析用に変更されたマジックセット法を用いることによって単純な前向き推論で実現できることを例題を用いて説明する．

例 4.1 マジックセット法の考え方に基づき変換された例題知識ベース $\overline{P}^{magic}_{ex}$（**図 6** 参照）を用いれば，単純な前向き推論を行うだけで問合せと無矛盾性制約との仮説依存関係を求めることができる．

知識ベース $\overline{P}_{ex}$（図 3 参照）において，例えば節番号 1 のルール $C_1 = m \leftarrow hp, hp, v.$ は，知識ベース $\overline{P}^{magic}_{ex}$ における 3 つのルール (1.1)，(1.2)，(1.3) に変換される．これらのルールの前向き推論による直観的意味は次のとおりである：

ルール (1.1)，(1.2)：ゴール $\leftarrow m$ に対応するアトム *magic_m* が与えられ，かつ，仮説 *hp*, *hp* を仮定する時，サブゴール $\leftarrow v$ を解くために *magic_v* が得られる．この時，サブゴールの導出において得られた候補仮説および，その導出時に用いられた入力節集合をゴールへ伝播するための述語 $cont_{11}(\{hp\}, \{\})$ も導かれる．ここで，述語 $cont_{11}$

ルール			
magic_m,hp,hp	→	magic_v.	(1.1)
magic_m,hp,hp	→	$cont_{11}$({hp},{}).	(1.2)
$cont_{11}$(H,Cls),v(Hv,Cls_v)	→	m(H∪Hv,{1}∪Cls∪Cls_v).	(1.3)
magic_d,hp,hp	→	magic_q.	(2.1)
magic_d,hp,hp	→	$cont_{21}$({hp},{}).	(2.2)
$cont_{21}$(H,Cls),q(Hq,Cls_q)	→	d(H∪Hq,{2}∪Cls∪Cls_q).	(2.3)
magic_v,r,hv	→	v({hv},{3}).	(3.1)
magic_q,l,hq	→	q({hq},{4}).	(4.1)
magic_s,r,hv	→	s({hv},{5}).	(5.1)
magic_s,l,hq	→	s({hq},{6}).	(6.1)
r. ns. l.			
仮説集合			
hp. hv. hq.			
無矛盾性制約			
magic_false₇,hp,nhp	→	false_p({hp},{7}).	(7.1)
magic_false₈	→	magic_s.	(8.1)
magic_false₈	→	$cont_{81}$({ }{}).	(8.2)
$cont_{81}$(H,Cls),s(Hs,Cls_s),ns	→	false₈(H∪Hq,{8}∪Cls∪Cls_s).	(8.3)
ゴール節			
	→	magic_m.	(9)
	→	magic_false₇.	(10)
	→	magic_false₈.	(11)

図 6 マジックセット法によって変換された例題知識ベース $\overline{P}^{magic}_{ex}$
Fig. 6 Magic sets transformed rules $\overline{P}^{magic}_{ex}$ of $\overline{P}_{ex}$.

* OLDT 反駁とマジックセット法等の対応については，例えば文献 14)，15) を参照されたい．

** O を問合せ，T_P を直接帰結オペレータ，I をエルブラン解釈とする．T_P の不動点計算における終了条件 $I = Tp(I)$ を $O' \in Tp(I)$ へ変更する（ただし，O' は O の例 (instance) とする）．

$(\{hp\}, \{\})$ の第一引数は，m の候補仮説の部分集合を表し，第二引数は，m の入力節集合の部分集合を表している．

ルール (1.3)： $cont_{11}(H, Cls)$ とともにゴール$\leftarrow v$ の解 $v(Hv, Cls_v)$ が得られた時，ゴール$\leftarrow m$ の解 $m(H \cup Hv, \{1\} \cup Cls \cup Cls_v)$ が得られる．ここで，述語 $m(H \cup Hv, \{1\} \cup Cls \cup Cls_v)$ の第一引数は，m の候補仮説を表し，第二引数は，m の入力節集合を表している（述語 $v(Hv, Cls_v)$ についても同様である）．

ルール(1.1)〜(1.3)に対する前向き推論は，節 C_1 に対する後向き推論に対応している．

図6の $\bar{P}^{magic}_{ex}$ に対して，前向き推論を行い，その不動点 $T\uparrow^{\omega}_{\bar{P}^{magic}_{ex}}$ を求める．その結果，ゴール(9)，(10)，(11)に対して以下の結果を得る．

$$T\uparrow^{\omega}_{\bar{P}^{magic}_{ex}} \supseteq \left\{ \begin{array}{l} m(\{hp, hv\}, \{1, 3\}). \\ false_8(\{hv\}, \{8, 5\}). \\ false_8(\{hq\}, \{8, 6\}). \end{array} \right\}$$

例えば，$m(\{hp, hv\}, \{1, 3\})$. の第一引数は，m の候補仮説，第二引数は，この候補仮説を導出するための入力節集合を示しており，この結果は，OLDT 反駁による $\bar{P}_{ex}$ の解析（図4参照）から得られる結果と等しい．□

4.2 入力節制約付マジックセット法

次に，入力節が制限されたOLDT反駁（例3.3 参照）と同様な計算をマジックセット法に基づいた前向き推論で実現することを考える．このために，通常のマジックセット法を次のように拡張する．

- ゴールに対する解代入と同時に，ゴールを証明するのに必要な仮説の集合を求める．
- マジック変換されたルールを用いた前向き推論は，プログラム解析によって得られた入力節集合によって制御されることにより，無駄な探索空間の枝刈りを行う．

例 4.2 知識ベース P_{ex}（図1参照）および，ゴール$\leftarrow m(b, Y, Z)$, $\leftarrow false_8$ を**図7**のようにマジック変換する．例えば述語 $magic\_m(X, Y, Z, Cls)$ の第四引数は，ゴール m の導出において選択すべき入力節集合を代入するための変数である．ここで，述語 $member(K, Cls)$ は，節番号 K が集合 Cls の要素であるか否かをチェックする述語である．このようにマジック変換された知識ベース P^{magic}_{ex} に対して，ゴール$\leftarrow false_8$ に対応する節番号(11)のゴール節を与えて前向き推論を行えば，図5と同様にしてゴール$\leftarrow m(b, Y, Z)$ の問合せに関する無駄な探索空間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，例2.1 の知識ベースを使用した．また，知識ベースに現れる事実の単位節の数を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8**に示す．実験は計算機 SUN 4/75 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，破線は文献 5) による手法の結果を示している．知識ベースの規模に関わらず，事前にプログラム解析を行うことにより，問合せ"$\leftarrow m(b, Y, Z)$"では約45%，問合せ"$\leftarrow d(b, Y, Z)$"では約80%程度に，推論時間がそれぞれ短縮されている．

また，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，

ルール

$$\begin{array}{ll}
magic_m(X,Y,Z,Cls), hp(X,s1), hp(Y,s2), member(1,Cls) & \\
\quad \rightarrow magic_v(Z,Cls). & (1.1) \\
magic_m(X,Y,Z,Cls), hp(X,s1), hp(Y,s2), member(1,Cls) & \\
\quad \rightarrow cont_{11}([X,Y,Z], \{hp(X,s1), hp(Y,s2)\}, Cls). & (1.2) \\
cont_{11}([X,Y,Z], H, Cls), v(Z,Hv,Cv), Cls \supseteq Cv & \\
\quad \rightarrow m(X,Y,Z,H \cup Hv, Cls). & (1.3) \\
 & \\
magic_d(X,Y,Z,Cls), hp(X,s1), hp(Y,s2), member(2,Cls) & \\
\quad \rightarrow magic_q(Z,Cls). & (2.1) \\
magic_d(X,Y,Z,Cls), hp(X,s1), hp(Y,s2), member(2,Cls) & \\
\quad \rightarrow cont_{21}([X,Y,Z], \{hp(X,s1), hp(Y,s2)\}, Cls). & (2.2) \\
cont_{21}([X,Y,Z], H, Cls), q(Z,Hq,Cq), Cls \supseteq Cq & \\
\quad \rightarrow d(X,Y,Z,H \cup Hq, Cls). & (2.3) \\
 & \\
magic_v(Z,Cls), r(Z), hv(Z), member(3,Cls) \rightarrow v(Z, \{hv(Z)\}, Cls). & (3.1) \\
magic_q(Z,Cls), l(Z), hq(Z), member(4,Cls) \rightarrow q(Z, \{hq(Z)\}, Cls). & (4.1) \\
magic_a(Z,Cls), r(Z), hv(Z), member(5,Cls) \rightarrow a(Z, \{hv(Z)\}, Cls). & (5.1) \\
magic_a(Z,Cls), l(Z), hq(Z), member(6,Cls) \rightarrow a(Z, \{hq(Z)\}, Cls). & (6.1)
\end{array}$$

r(101). r(102).　l(201). l(202). l(203). l(204).　na(101). na(204).

仮説集合

hp(b,s1). hp(c,s1). hp(e,s2). hp(f,s2).　hv(Z).　hq(Z).

無矛盾性制約

$$\begin{array}{ll}
magic_false_7([], Cls), hp(X,Y), nhp(X,Y), member(7,Cls) & \\
\quad \rightarrow false_7([], \{hp(X,Y)\}, Cls). & (7.1) \\
magic_false_8([], Cls), member(8,Cls) \rightarrow magic_a(Z,Cls). & (8.1) \\
magic_false_8([], Cls), member(8,Cls) \rightarrow cont_{81}, ([], \{\}, Cls). & (8.2) \\
cont_{81}([], H, Cls), a(Z,Ha,Ca), na(Z), Cls \supseteq Ca & \\
\quad \rightarrow false_8([], H \cup Ha, Cls) & (8.3)
\end{array}$$

ゴール節

$$\begin{array}{ll}
\rightarrow magic_m(b, Y, Z, \{1,3\}). & (9) \\
\rightarrow magic_false_8([], \{8,5\}). & (11)
\end{array}$$

図 7 マジックセット法によって変換された例題知識ベース P^{magic}_{ex}
Fig. 7 Magic sets transformed rules P^{magic}_{ex} of P_{ex}.

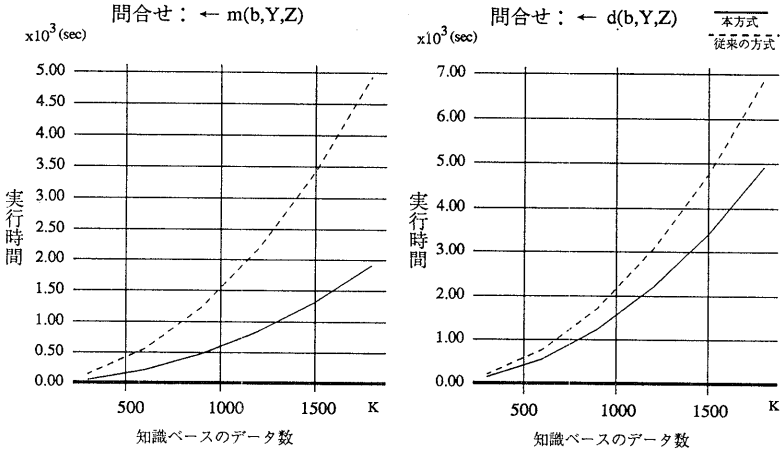


図 8 実験結果
Fig. 8 Empirical results.

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

一階述語論理を用いた仮説推論の高速化に対して，プログラム解析に基づく無矛盾性の検査の効率化の提案を行った．本論文で提案したプログラム解析法は，ホーン節述語論理で表現された知識ベースを命題論理のレベルに抽象化することにより，与えられた問合せと知識ベースの論理的制約に対して，実際の推論において評価すべき探索空間を，近似的に求めるものである．この解析法を用いて，仮説推論における制約節の評価時において，無駄な探索空間の絞り込みを行うような，また，マジックセット法に基づきつつ，それをプログラム解析結果を利用できるような形に拡張した方式により仮説推論システムを実装しその有効性を確認した．本論文で述べたマジックセット法に基づく推論システムの実装法は，それが前向き推論の一制御法を提案している点（例えば文献 16)）から，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．

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．同 WG の委員諸氏に深謝いたします．

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) Stickel, M. E.: Upside-Down Meta-Interpretation of the Model Elimination Theorem-Prover Procedure for Deduction and Abduction, Technical Report TR-664, ICOT (1991).
5) Ohta, Y. and Inoue, K.: A Forward-Chaining Hypothetical Reasoner Based on Upside-Down Meta-Interpretation, *Proc. of the International Conference on FGCS '92*, pp. 522-529 (1992).
6) Kondo, A., Makino, T. and Ishizuka, M.: An Efficient Hypothetical Reasoing System for Predicate-logic Knowledge-base, *Proc. IEEE, Intl. Conf. on Tools for AI*, pp. 60-67, San Jose (1991).
7) Bancilhon, F., Maier, D., Sagiv, U. and Ullman, J. D.: Magic Sets and Other Strange Ways to Implement Logic Programs, *Proc. Fifth ACM SIGMOD-SIGACT-SIGART Symp. on Principles of Database Systems*, pp. 1-15 (1986).
8) Vieille, L.: Recursive Axioms in Deductive Databases: The Query/Subquery Approach, *Proc. First Intl. Conf. on Expert Database Systems*, pp. 179-193, Charleston (1986).
9) Kato, S., Seki, H. and Itoh, H.: An Efficient Abductive Reasoning System Based on Program Analysis, *Static Analysis, Proc. of the 3rd Intl. Workshop, Lecture Notes in Computer Science*, Cousot, P. et al. (eds.), Vol. 724, pp.

230-241, Springer-Verlag, Padova (1993).

10) Kato, S., Seki, H. and Itoh, H.: A Deductive Database Approach to Abductive Reasoning, *Proc. of the 3rd Intl. Symposium on Next Generation Database Systems and Their Applications*, pp. 77-84, Fukuoka (1993).

11) 太田好彦, 井上克巳: ATMSを用いた前向き仮説推論システムにおける効率的な推論方式, 人工知能学会誌, Vol. 6, No. 2, pp. 247-259 (1991).

12) Tamaki, H. and Sato, T.: OLD Resolution with Tabulation, *Proc. of the 3rd ICLP*, pp. 84-98, London (1986).

13) Ng, H.T. and Mooney, R.J.: An Efficient First-Order Horn-Clause Abduction System Based on the ATMS, *Proc. of the AAAI-91*, pp. 494-499 (1991).

14) Seki, H.: On the Power of Alexander Templates, *Proc. Eighth ACM SIGMOD-SIGACT-SIGART Symp. on Principles of Database Systems*, pp. 150-159 (1989).

15) 宮崎収兄, 世木博久: 演繹データベースの問合せ処理, 情報処理, Vol. 31, No. 2, pp. 216-224 (1990).

16) Ramakrishnan, R., Srivastava, D. and Sudarshan, S.: Controlling the Search in Bottom-up Evaluation, *Joint International Conference and Symposium on Logic Programming*, pp. 273-287 (1992).

付　　録

命題3.1を証明するために，まず次の補題を示す．

A 補題 3.1

補題 3.1 P をプログラム，$\leftarrow G$ をゴールとする．$\Delta, \mathcal{H}_g$ および Ξ を **M** 演算子付アトムの連言，Γ および Λ をアトムの連言とする．$N_1=\leftarrow\Delta, g, \Gamma$ を $P\cup\{\leftarrow G\}$ の OLDT 反駁木におけるゴールとし，g を選択されたアトムとする．更に，以下の条件が成り立つと仮定する．

- N_1 の反駁における$\leftarrow g$ の部分反駁で，$\leftarrow\mathcal{H}_g$ を成功葉として持つものが存在する．
- $\bar{P}\cup\{\leftarrow\bar{G}\}$ の OLDT 反駁木においてノード $N_2=\leftarrow\Xi, \bar{g}, \Lambda$ で，$\bar{g}$ を選択されたアトムとするものが存在する．

このとき，N_2 の反駁における$\leftarrow\bar{g}$ の部分反駁で，$\leftarrow\overline{\mathcal{H}}_{\bar{g}}$ を成功葉として持つものが必ず存在する．□

証明：部分反駁の長さに関する帰納法で証明する．ここでは，参照 (lookup) ノードの解テーブル参照による部分反駁の長さは，対応する解ノードに対する部分反駁の長さに等しいとする．

1. $\leftarrow g$ の部分反駁の長さが1のとき，
$^\exists g_0\leftarrow\in P$ かつ $^\exists\bar{g}_0\leftarrow\in\bar{P}$ (g_0 は g と単一化可能なアトム) であるので，N_1 および N_2 に対する一段階の OLDT 導出を考えれば明らかである．
2. $\leftarrow g$ の部分反駁の長さが $k+1$ のとき，
$P\cup\{\leftarrow G\}$ の OLDT 反駁木に現れるすべてのゴールに対して，その部分反駁の長さが k 以下の場合については補題が成り立つと仮定する．

場合 a：N_1 が解ノードのとき，
$^\exists g_0\leftarrow A_1, \cdots, A_n\in P$ より，$\leftarrow g$ の部分反駁木は，$\leftarrow g$ のサブゴールとして $\leftarrow A_1\theta, \cdots, A_n\theta$ ($\theta=mgu(g, g_0)$) を持つ．$\leftarrow g$ の部分反駁は$\leftarrow A_i\theta$ ($1\le i\le n$) の部分反駁を繋げたものであり，$\leftarrow A_i\theta$ の部分反駁の長さはすべて k 以下である．一方，$\bar{P}\cup\{\leftarrow\bar{G}\}$ の OLDT 反駁において，次の2つの場合を考える．

場合 a-i：N_2 が解ノードのとき，
$^\exists\bar{g}_0\leftarrow\bar{A}_1, \cdots, \bar{A}_n\in\bar{P}$ より，N_2 はサブゴール $\leftarrow\Xi, \bar{A}_1, \cdots, \bar{A}_n, \Lambda$ を持つ．ここで，$\leftarrow\bar{A}_i$ ($1\le i\le n$) の部分反駁については，帰納法の仮定より補題が成立する．従って，$\leftarrow\bar{g}$ の部分反駁で，$\leftarrow\overline{\mathcal{H}}_{\bar{g}}$ を成功葉として持つものが存在する．

場合 a-ii：N_2 が参照ノードのとき，
N_2 に対応する解ノードについては場合 a-i と同様のことが成り立つので，N_2 についても上の場合と同様である．

場合 b：N_1 が参照ノードのとき，
N_1 に対応する解ノードの部分反駁の長さは $k+1$ であるので，場合 a と同様のことが成り立つ．

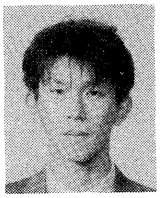
以上より，部分反駁の長さが $k+1$ でも補題は成立する．□

B 命題 3.1 の証明

(i)の証明：定義2.1より，F 自身が矛盾することはない．すなわち $F\nvdash\square$ である．従って，H_0 が条件 (AR 2) を満たさないとき，つまり，$F\cup H_0\vdash\square$ となるときは，$F\cup H_0\cup\{\leftarrow false\}$ の OLDT 反駁木の成功葉には必ず仮説集合が現れ，任意の候補仮説 H_{false} は H_0 の部分集合となる．ここで，$P\supseteq F\cup H_0$ より，$P\cup\{\leftarrow false\}$ の反駁において (H_0, H_{false}) が O に対する矛盾仮説対となる H_{false} が必ず存在する．

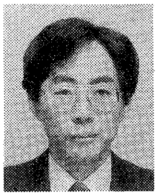
(ii)の証明：補題1より，$P\cup\{\leftarrow false\}$ および $\bar{P}\cup\{\leftarrow false\}$ の OLDT 反駁木について考えれば明らか．□

(平成6年3月11日受付)
(平成6年6月20日採録)



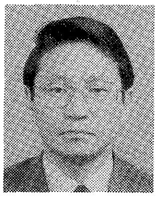
加藤 昇平（学生会員）

1993年名古屋工業大学電気情報工学科卒業．現在同大学院工学研究科博士前期課程在学中．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工学部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．