命題レベルの高速解法の利用を図る コストに基づく述語論理版仮説推論法

A Predicate-logic-version Mehtod for Cost-based Hypothetical Reasoning employing Efficient Propositional-level Reasoning Mechanism

棚橋　一也 *,†1　　福田　茂紀 *,†2　　石塚　　満 *
Kazuya Tanahashi　　Shigeki Fukuta　　Mitsuru Ishizuka

* 東京大学 工学部 電子情報工学科
Dept. of Information and Communication Engineering, School of Engineering, The University of Tokyo

19YY 年 MM 月 DD 日　受理

Keywords: cost-based hypothetical reasoning, efficient abductive reasoning, propositional/predicate-level reasoning, reformation of predicate rules.

Summary

For propositional-version hypothetical reasoning (or abduction), particularly for its cost-based one, some efficient methods have been developed, which achieve polynomial-time reasoning with respect to problem size to compute a near-optimal solution. However, for predicate-version hypothetical reasoning that allows rich and compact knowledge representation, it seems difficult to find such a method with polynomial-order efficiency as long as we stick to symbolic manipulation. As a result, there has been no good efficient method so far in predicate-logic domain.

A naive approach for efficient predicate-version hypothetical reasoning is to transform predicate knowledge into propositional knowledge, and then to apply an efficient method in propositional domain. This can be accomplished by a transformation in the Herbrand universe; however, this is impractical since a vast number of propositional clauses will be produced. We tried to use a deductive database technique, namely, QSQR method, to extract a knowledge portion being related to the proof of a given goal of hypothetical reasoning, and then to transform only this portion into propositional knowledge. Nevertheless, it can not improve reasoning efficiency enough because a considerable computational time is required for the transformation into propositional knowledge by the QSQR method, and the number of the resulting propositional clauses is still large. In this paper, we propose a knowledge reformation scheme for predicate rules such that the transformation into propositional rules is efficient and the number of the resulting propositional rules is small. This knowledge reformation is based on unfolding/folding processes for predicate rules. Although its applicable range is still limited, it is shown experimentally that, by introducing this knowledge reformation, a large amount of efficiency improvement is achieved for predicate-version cost-based hypothetical reasoning

1. まえがき

仮説推論 [石塚 96] は知識処理の有用な枠組みであるが、仮説推論自体はNP 完全、あるいはNP 困難問題であり [Kautz89, Bylander89]、実用問題に適用するには低い推論速度が大きな問題であった [石塚 94]。しかし、命題版仮説推論に関してはいくつかの有効な高速推論メカニズムが考案、開発されてきている。特に、コストに基づく仮説推論（あるいはコストに基づくアブダクション [Charniak90]）の準最適解計算については、数理計画の手法も利用することで、問題サイズに対して低次の多項式時間となる高速化が達成されている [岡本 93, 大澤 94,95, Ohsawa97, 松尾 98]。命題レベルの推論に

---

†1　現在、(株) 富士通
†2　現在、(株) 富士通研究所

ついては仮説推論に限らず、SAT(Satisfiability Problem) や CSP(Constraint Satisfaction Problem) についても GSAT[Selman93a, 93b] や Gu の方法 [Gu93] のように、効果的な方法が考案されている。命題レベルの推論は多くの AI 問題の基盤となる [Selman97]。

一方、命題論理変数と論理記号による命題論理は知識表現能力の点では必ずしも十分でなく、変数を含む項を引数としてとる述語表現の(1階)述語論理が必要になることが多い。端的には、少数の述語論理式で表わされる知識でも、命題論理式では数100、数1000倍もの数になることも稀ではない。

述語論理は表現力は高いものの、効率的推論の実現に関しては難しくなる。関数を含まない1階述語ホーン節表現と無矛盾性制約を扱う仮説推論については、[近藤 93,94, Kondo93,96] のような効率的推論法が開発されている。しかしこれは、問題規模に対して指数的に増大する推論速度の壁を克服するものとはなっていない。大澤らは彼等のネットワーク化バブル伝播法(NBP 法)を述語の引数間に適用することにより、(関数なし)述語ホーン節表現のコストに基づく仮説推論に対しても、指数オーダの推論時間の壁を克服するような方法を開発している [大澤 95, Ohsawa97]。しかし、メカニズムは大幅に複雑化し、所要メモリ量の増大などで、小規模問題の範囲にしか使用できず、スケールアップも難しい。

命題論理レベルの良い解法が存在するなら、述語論理レベルの問題を命題論理レベルに展開して、命題論理レベルの高速解法の利用を図るというのは普通に考えつくことであろう。述語論理の変数はエルブラン領域に展開すれば、定数(基礎項)とすることができるので、述語論理表現を命題論理表現へと原理的には展開できることになる。(関数なしの場合は有限個の命題論理式となる。)しかし、このような素朴な方法は生成される命題論理の節数が膨大なものとなり、実用的には使えない。

我々は以前に、演繹データベースの問い合わせ処理手法である QSQR 法 [Vieille86, 西尾 88] を用いて、述語論理版仮説推論においてゴールの証明に関与する部分のみを命題論理に展開する方法を試みた [福田 97]。そして、生成されたコストに基づく命題論理版仮説推論問題に対して、準最適解を多項式時間で求めることができるネットワーク化バブル伝播法の適用を試みた。しかし、これでも命題論理への展開時間がかかり、生成される命題節数も多くなり過ぎ、十分な高速化の効果が得られなかった。

本論文では、述語論理版のコストに基づく仮説推論問題を命題レベルに展開して準最適解を計算するアプローチにおいて、命題への展開時間を短縮し生成される命題節数を抑制するように、元の述語ホーン節を変型する操作(リフォメーション)を導入した手法を示し、それによって得られる高速化の効果を示す。すなわち、本論文のコストに基づく述語論理版仮説推論システムでは、図1に示すように、命題論理式への変換が効率的に行えるように元の述語ホーン節(ルール)のリフォメーションを行い、QSQR 法によりゴールの証明に関与する部分のみを抽出して命題化し、命題論理版仮説推論問題に変換する。そして、高速の命題論理版仮説推論法(具体的にはネットワーク化バブル伝播法 [大澤 94,95, Ohsawa97]、及び SL 法 [松尾 98])を適用し、準最適解を求める。

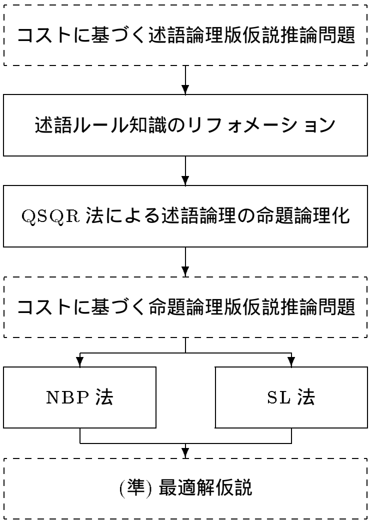

図1 システムの基本構成

論理表現や論理プログラムの変型や等価変換に関する研究はこれまでにもあったが [古川 87, 堂前 94]、述語表現の命題化を効率的に行うための述語論理知識の変型という観点からの研究はこれまでに見当たらない。

2. 背景となる手法

最初に本論文の手法の背景となる手法について簡単に記しておく。

2・1 コストに基づく仮説推論

仮説推論の知識ベースには、対象世界で常に成り立つ背景知識 Σ と、常に成り立つとは限らず他と矛盾する可能性を有する仮説知識の集合 H を記述する。H

の部分集合をh、すなわち$h \subseteq H$とする。仮説推論の基本動作は次の通りである。ゴールGが与えられた時、まずΣからGが演繹的に証明できるか確かめる。Σからだけでは証明できない時、次の条件を満たすGを証明（説明）するのに必要な解仮説hを求める。

(1)　$\Sigma \cup h \vdash G$　（ΣとhからGが証明できる；$\vdash$は演繹的証明を示す論理記号）

(2)　$\Sigma \cup h$は無矛盾

すなわち、背景知識ΣだけからGを証明できればそれでよいが、ΣだけではGを証明できないとき、仮説集合Hの中から無矛盾な部分集合hを切り出し、Σと合わせてGを証明するようにする。多くの場合、hは極小性（または非冗長性）、すなわちhの真部分集合$h'(\subset h)$が上記の条件を満たさないことも要請される。さらに、個々の要素仮説に重みが付されているとき、含まれる要素仮説の重みの和をコストと定義すると、コストが最小の解仮説hを求めたい場合を重み付き仮説推論、あるいはコストに基づく仮説推論という[石塚96, Charniak90]。

本論文で対象とするのは、関数を含まない1階述語ホーン節と無矛盾性制約で表されるコストに基づく仮説推論の問題である。

2・2　ネットワーク化バブル伝播法（NBP法）

コストに基づく命題版仮説推論の準最適解の多項式時間計算法としては、最初に0-1整数計画の掃出し補数法の適用を図った岡本らの方法がある[岡本93]。ネットワーク化バブル伝播（NBP）法[大澤94,95, Ohsawa97]は、この掃出し補数法の動作を知識ネットワーク上の動作に置き換え、同時に問題の知識構造を利用することにより推論の効率化を図ったものである。コストに基づく命題版仮説推論については、可能な要素仮説数をNとすると、準最適解をほぼ$N^{1.5}$の多項式時間で見出すという非常に高い効率を達成している。

NBP法はコストに基づく述語論理版仮説推論に対しても適用が図られている。これは述語引数である項の間にネットワークを形成し、巧妙な方法でネットワーク化バブル伝播法による推論を可能としたものである。しかし、メカニズムの複雑化に伴い、実用規模の問題への適用へのスケールアップが難しい。

2・3　Slide-down & Lift-up法（SL法）

掃出し補数法、NBP法に共通するのは、記号推論問題を線形不等式制約に変換し、単体法により実数最適解（各変数は0 ~ 1間の値をとり真/偽、あるいは1/0の2値ではない）を効率的に求め、これを探索の初期点に設定し、この近傍の1/0準最適解（各変数が真/偽の値となる）を局所探索により求めるというものである。この場合の局所探索は、メカニズムが複雑であり、把握しにくくなってしまっている。SL法[松尾98]はこの局所探索部を、非線形計画法の手法（具体的には非線形関数の最急降下法）の利用と、局所最適点への捕獲から脱出機構として、局所的な不整合状態を正すような変数の真/偽状態への固定化を導入して実現したものである。これによって、この局所探索部の動作が簡潔になり、把握しやすくなっている。コストに基づく仮説推論の準最適解の計算時間は、現状ではNBP法にやや劣り、約$N^{1.85}$の多項式時間となっているが、所要メモリ量はNBP法よりも大幅に少ない。

2・4　QSQR法

演繹データベースの効率的問い合わせ処理法としてよく知られているのが、マジックセット法であるが、QSQR法[Vieille86, 西尾88]はトップダウンの制御のもとで、ボトムアップの解の合成を行うことで、マジックセット法と同等の効率を達成する。対象は関数を含まない述語ホーン節の知識（演繹データベースのDatalog表現）である。QSQR法は証明木のある分枝からボトムアップに生成された変数への束縛情報の横方向（兄弟ノード）への伝播と、ボトムアップに生成した部分解をlemmaとして保持して再計算を回避することで、効率的なゴール指向ボトムアップ推論を可能とする。

本研究では、ゴールの証明に関係する部分の述語ホーン節のみを効率的に抽出するためにQSQR法を用いる。そして、抽出された基礎具体化されたリテラルから成る述語ルールを命題ルールに変換する。ここで基礎具体化されたリテラルから成る述語ルールが生成されるのは、次章で記すように仮説として基礎具体化された単節の要素仮説のみを扱い、QSQR法によりボトムアップにゴールの証明に関係する知識を構成することによる。

QSQR法が対象とする演繹データベースで用いられる用語について、以下に若干の説明を記しておく。

- EDB (extensional database): 論理プログラムのファクト集合
- IDB (intensional database): ルールの集合であるが、通常本体に負のリテラルがない確定節
- EDB述語: EDB中に定義されている述語
- IDB述語: IDB中のルールを定義するために導入された述語

3. 述語論理版仮説推論問題から命題版への変換

効率的な命題版仮説推論法（NBP 法や SL 法）を利用して、述語論理版仮説推論法を構成するための変換法を考える。原理的には関数を含まない述語論理式はエルブラン領域に展開することで命題論理化できるが、これは生成される命題節数が膨大なものとなり実用的ではない。

ここでは、ゴールの証明（説明）に関係する部分のみを抽出して命題論理化を図るために、QSQR 法を用いる。ここで、コストに基づく述語論理版仮説推論は次の形式で表されたものを対象とする。

- 関数なし述語ホーン節（背景知識）
- 矛盾の組合せを表す制約（矛盾制約）
- 数値的重み（ディフォルト値は 1）が付与され、基礎具体化（定数化）された単節の要素仮説

（要素仮説についてはルール型知識も仮説として扱うことが必要になることもあるが、これは単節の要素仮説とホーン節背景知識に変換可能であることより、上記の要素仮説の扱いは適用範囲を狭めるものとはならない。）

命題論理化を図るには、QSQR 法で矛盾制約は無視してゴール（基礎具体化された単節、あるいはそのような単節の連言で与えられる）の証明を行うことにより、ゴールの証明に関係する基礎具体化された単節の要素仮説と、これも基礎具体化された述語ホーン節を抽出する。そして、これらを命題記号、及び命題ホーン節へ変換する。矛盾制約についても基礎具体化された要素仮説の組合せに展開し、命題化された矛盾制約へと変換する。この時、上記でゴールの証明に関係するとして抽出された要素仮説のみの連言で構成される矛盾制約だけに制限することにより、扱う矛盾制約を与えられたゴールの証明に関係するものに限定し、不要に数が増えることを避けることができる。

以上のようにして、ゴールの証明 (説明) に関与する部分のみを命題版仮説推論問題へと変換でき、高速命題版仮説推論手法により準最適解を計算できることになる [福田 97]。しかしゴールの証明に関与する部分に制限しても、命題への展開は基礎具体化された述語リテラル（これが命題変数になる）を用い可能性のある証明木をすべて構成することになり、しばしば規模が非常に大きくなり、命題への展開時間、および生成される命題ルール（ホーン節）の数が増大してしまうという問題を生じる。この結果、期待する推論速度の向上が達成されないことがしばしば生じる。

この課題に対処するため、本研究では命題への展開を効率化し、生成される命題ルール (ホーン節) 数を抑制するような、元来の述語ホーン節に変型 (リフォメーション) を施す操作を導入する。

以下に適用している unfolding フェーズ（圧縮フェーズ）、第 1 folding フェーズ（非孤立変数分離フェーズ）、第 2 folding フェーズ（孤立変数分離フェーズ）について順に述べる。なお、ここでのリフォメーションはルール型知識（演繹データベースの IDB 述語）のみを対象として行い、単節（EDB 述語）の事実及び仮説を用いての展開は QSQR 法によって行うものとする。

Prolog 型ルール

$$p(X,Z) :- q1(X,Y), q2(Y,A), q3(Z).$$

において各種変数と操作の定義を示す。

頭部変数　頭部の引数となっている変数（上の例で X,Z）

非頭部変数　頭部変数以外の変数

孤立変数　非頭部変数で、ルール本体で、一つの述語の中にしか現れない変数（上の例で A）

非孤立変数　非頭部変数で、孤立変数でない変数（上の例で Y）

unfolding　2 つのルール A、B があり、B の頭部とユニファイ可能な A の本体の述語リテラルをユニフィケーションした B の本体で置き換える。置き換える対象が複数あるときには、左優先で一つだけ置き換え、他はそのままにする。

3・1 unfolding フェーズ（圧縮フェーズ）

次のような知識ベースの状況を考える。

```
[I-i]  g(X,Z):- p(X,Y),q(Y,Z).
       q(Y,Z):- r(Y),s(Z).
[I-ii] g(X,Z):- p(X,Y),r(Y),s(Z).
```

[I-ii] は [I-i] の第 1 式に第 2 式を代入し unfolding 操作を行ったものである。ここで、各変数 X, Y, Z は 3 種の定数値 a, b, c をとる可能性があるとする。g(X,Y) をゴールとしてゴールの証明に関係する部分のみ述語ルールの命題化を図る場合、まず生成される命題ルール数を指標にして考えてみる。（一般にこの数が少なければ命題への展開時間も少ない。）[I-i] の第 1 式は $3 \times 3 \times 3 = 27$（非孤立変数 Y は同一の定数となる必要があるため）、第 2 式は $3 \times 3 = 9$、合計 36 個の命題ルールに変換されることになる。一方、[I-ii] では、非孤立変数 Y は同一定数になる必要があることから $3 \times 3 \times 3 = 27$ 個の命題ルールによる変換できることになる。ゴールが g(a,b) のような場合は、命題ルー

ル数は [I-i] では 3 + 3 = 6 個、[I-ii] では 3 個となる。このように unfolding により生成される命題ルール数を削減できる可能性がある。

従って、このフェーズではこのような述語ルールの unfolding 処理を行う。ただし、次のような再帰ルール

$$\mathrm{g(X):-p(X,Y),g(Y).}$$

の再帰リテラル g(X) については、これを unfolding し、

$$\mathrm{g(X):-p(X,Y),p(Y,Y2),g(Y2).}$$

にするようなことはしない。

3・2 第 1 folding フェーズ（非孤立変数分離フェーズ）

上記のような unfolding 処理が生成される命題ルール数の削減の点で有効であるとは必ずしも限らない。1 つのルール中に 2 個の非孤立変数を含む次のような知識ベースの状況を考えてみる。

```
[II-i]  g(X,Z):- p1(X,Y1),p2(Y1,Y2),
                 p3(Y2,Z).
[II-ii] g(X,Z):- p1(X,Y1),q(Y1,Z).
        q(Y1,Z):- p2(Y1,Y2),p3(Y2,Z).
```

このとき [II-i] は [II-ii] の第 2 式を第 1 式に代入し unfolding した関係になっている。状況を簡単にし、X,Y1,Y2,Z の変数は各々a,b,c,$\cdots$, のような n 種の定数をとる可能性があるとする。g(X,Z) をゴールとしてゴールの証明に関係する部分のみの述語ルールを命題化するとすると、[II-i] では n^4 個の命題ルールに、[II-ii] では $2 \times n^3$ 個の命題ルールに変換されることになる。この場合、[II-i] は [II-ii] の $n/2$ 倍の命題ルール数となり、n が大きな程 [II-ii] の方が有利になる。別途次節で例示するように、生成される命題ルール数の減少に対応して、命題への展開時間も減少する。

従って、このフェーズでは 1 述語ルール中に分離可能な複数個の非孤立変数が存在する時、これを複数の述語ルールに分離する。ここで分離可能とは、以下に示す例のように新述語の導入により、非孤立変数が別の述語ルールに現れるようにすることができる場合を指す。

$$\mathrm{g(X,Z):-p1(X,Y1),p2(Y1,Y2),p3(Y2,Z).}$$

の場合、新述語 new(X,Y) を導入し、

$$\mathrm{g(X,Z):-new(X,Y2),p3(Y2,Z).}$$

$$\mathrm{new(X,Y2):-p1(X,Y1),p2(Y1,Y2).}$$

のように、元の 1 述語ルール中に現れていた非孤立変数 Y1 と Y2 とを、別の述語ルールに分離する。

ここでの非孤立変数分離は、対象のルール本体を左優先で見ていき、分離可能な非孤立変数を含むリテラルを順次分離していくようにしている。分離したリテラルは新述語を頭部とする新ルールを作成するのだが、既に存在するルールと同一になる時は新ルールの作成は行わないようにする（Prolog で通常言われているところの folding 処理に相当）。

複数個の非孤立変数を含むルールであっても、以下の例のようにそれらが分離不可能である場合には、このリフォメーション処理は行わない。

$$\mathrm{g(X,Z):-p1(X,Y1,Y2),p2(Y1,Y2,Z).}$$

3・3 第 2 folding フェーズ（孤立変数分離フェーズ）

複数個（m とする）の定数値をとる可能性を持つ孤立変数はそれが含まれる述語ルールが命題ルール化される時に、m 倍の数にしてしまうことになる。これは孤立変数だけを含む述語ルールに分離することにより回避することができる。

例えば、

$$\mathrm{g(X,Y):-p1(X,Z),p2(Z,Y),p3(A).}$$

（A が孤立変数）

に対して、新アトムを導入し、次のように分離する。

$$\mathrm{g(X,Y):-p1(X,Z),p2(Z,Y),new1.}$$

$$\mathrm{new1:-p3(A).}$$

この第 1 式から k 個の命題ルールが生成され、第 2 式が m 個の命題ルールが生成される（即ち A が m 個の定数値をとる）時、$k+m$ 個の命題ルール数となるが、元の形式のままであると $k \times m$ 個の数になってしまう。

従って、このフェーズでは新アトム導入による孤立変数分離の folding 処理を行う。p3(Z,A)｛A は孤立変数だが、Z は孤立変数でない｝のようなリテラルの分離はできないが、p3(a,A)｛a は定数｝のように孤立変数以外の変数を含まないリテラルは分離する。

3・4 各リフォメーションフェーズの適用順序

以上のリフォメーション処理を次の順序で適用している。

(1) unfolding フェーズ（圧縮フェーズ）
(2) 第 1 folding フェーズ（非孤立変数分離フェーズ）
(3) 第 2 folding フェーズ（孤立変数分離フェーズ）

この順序は、まず可能な unfolding を行って連結した述語ルールを生成し、その後に非孤立変数分離及び孤

立変数分離の folding により、ルールの有効な分離を行うことになる。

この述語レベルのリフォメーションは矛盾制約知識(ルール)にも適用され、その後に命題の矛盾制約知識へと展開される。

なお、孤立変数分離の第 2 folding フェーズを除き、他の述語ルール・リフォメーション処理の多様な知識表現状況に対しての生成される命題ルール数削減量の理論的検討は今後の課題であるが、上記の (1),(2),(3) の処理順序により、3.1, 3.2, 3.3 節で示した削減効果が最も有効に発揮されるようになる。次章で実験的検討により、このリフォメーション処理が良好な結果をもたらすことを示す。

4. リフォメーションの効果

ここでは上記の述語知識のリフォメーションが、コストに基づく述語版仮説推論の準最適解計算の全推論時間の効率化を実現する効果を、例によって示す。

図 2は double_loop(a,b) をゴールとして与えると、X=a から Y1,Y2,Y3,Y4(リンク 1、リンク 2、リンク 3、リンク 4、リンク 5) を経由して X=a に戻るループと、同様に Z=b から Y1,Y2,Y3,Y4 (リンク 1、リンク 2、リンク 3、リンク 4、リンク 5) を経由して Z=b に戻るループの 2 本のループ経路を、矛盾制約に違反せず使用する各リンクの重みの和 (コスト) が最も小さい条件で見い出そうとする。図 2は Y1 ~ Y4 が各々5 個 (a0 ~ a4) の値をとれ、Y1 と Y2 間のような各リンクは $5 \times 5 = 25$ 本の可能性を持つ場合であるが、Y1 ~ Y4 のとり得る値と存在するリンク数を変えることにより問題規模を変えることができる。(図 2は見かけは要素仮説 125 の場合となるが、2 本のループ各々でこの数だけの仮説を持つ構造となることから、実質は要素仮説数 250 の場合に相当することになる。)

図 2の述語ルールのリフォメーションでは、unfolding フェーズで一度

```
double_loop(X,Z):-
   link1(X,Y1),link2(Y1,Y2),link3(Y2,Y3),
   link4(Y3,Y4),link5(Y4,X),
   link1(Z,YY1),link2(YY1,YY2),
   link3(YY2,YY3),link4(YY3,YY4),
   link5(YY4,Z).
```

が生成され、これが第 1 folding フェーズで複数ルールに折り畳まれる。この時、本体の左から分離可能な非孤立変数が分離され新ルールが生成されていくのだが YY1 以降の非孤立変数の分離時には既に対応するルールが存在するので、新ルールは生成されないことになる (通常用いられる意味での folding 処理に相当)。

```
/***** 元の知識ベース (リフォメーション前) *****/
double_loop(X,Z) :- loop(X),loop(Z).
loop(X) :- link1(X,Y1),link2(Y1,Y2),link3(Y2,Y3),
             link4(Y3,Y4),link5(Y4,X).
inc :- link1(X,Y),link2(Y,X).      /* 矛盾制約
inc :- link2(X,Y),link3(Y,X).      /* 矛盾制約
inc :- link3(X,Y),link4(Y,X).      /* 矛盾制約
inc :- link4(X,Y),link5(Y,X).      /* 矛盾制約
inc :- link5(X,Y),link1(Y,X).      /* 矛盾制約
/* 以下は重み付き (/の後の数値) 要素仮説
link1(a0,a0)/3.
link1(a0,a1)/9.
link1(a0,a2)/1.
link1(a0,a3)/5.
link1(a0,a4)/2.
link1(a1,a0)/10.
link1(a1,a1)/3.
link1(a1,a2)/5.
link1(a1,a3)/10.
link1(a1,a4)/10.
link1(a2,a0)/9.
link1(a2,a1)/9.
link1(a2,a2)/1.
link1(a2,a3)/5.
link1(a2,a4)/10.
link1(a3,a0)/5.
link1(a3,a1)/4.
link1(a3,a2)/3.
link1(a3,a3)/7.
link1(a3,a4)/3.
link1(a4,a0)/2.
link1(a4,a1)/10.
link1(a4,a2)/5.
link1(a4,a3)/2.
link1(a4,a4)/4.
/* 以下、link2,link3,link4,link5 についての要素仮説
/* も同様で省略

/***** リフォメーション後の知識ベース *****/
double_loop(X,Z) :- nn4(X),nn3(Z,XX6),link5(XX6,Z).
nn4(X) :- nn3(X,Y4),link5(Y4,X).
nn3(X,Y4) :- nn2(X,Y3),link4(Y3,Y4).
nn2(X,Y3) :- nn1(X,Y2),link3(Y2,Y3).
nn1(X,Z) :- link1(X,Y1),link2(Y1,Y2).
/* 矛盾制約と要素仮説については変化なしで同じ
```

図 2　サンプル知識ベース 2 (リフォメーションの前と後)

図 3はリフォメーション前後の述語知識を展開して生成される命題ルール数の比を示している。問題規模の増大につれて、生成される命題ルール数のリフォメーションによる減少率は大になっている。

図 4は命題ルールへの展開を行い、コストに基づく

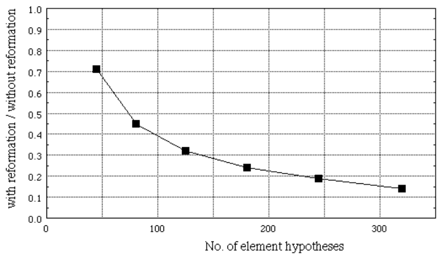

図 3　リフォメーションによる生成命題ルール数の減少効果（図 2の知識ベースのとき）

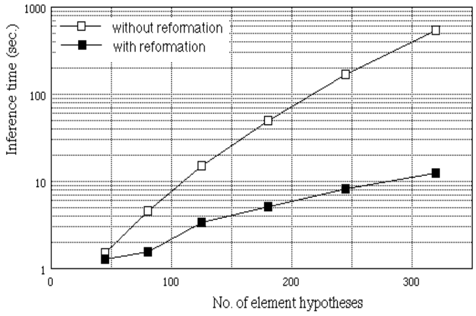

図 4　リフォメーションによる推論時間の高速化効果（図 2の知識ベースのとき）

命題版仮説推論の高速解法であるネットワーク化バブル伝播法 (NBP 法) を適用して、準最適解仮説を求めるまでの推論時間を示している。リフォメーションの効果は大変大であり、問題規模の増大につれて大きくなっていく。なお、推論時間のうち 90%程は QSQR 法による述語ルールの命題ルールへの展開が占めている。QSQR 法に代わる手法によるこの展開時間の縮小は今後の課題であるが、ここでのリフォメーションはこの展開時間の縮小にも寄与している。命題ルール数と使用される命題変数の数 (N とする) は比例すると仮定すると、命題レベルの準最適解計算はネットワーク化バブル伝播法でほぼ $O(N^{1.5})$ の推論時間であるので、命題ルール数が $1/n$ になると $1/n^{1.5}$ に短縮されることとなる。

KICK-HOPE II [近藤 94, Kondo96] は述語論理版仮説推論の最適解を述語表現のまま求めるシステムであるが、この推論時間は図 4 のリフォメーションなしの場合の 2,3 倍程度となっている。(この KICK-HOPE II もリフォメーションした述語知識を対象にすると効率化が図れる可能性があるが、未検討である。)

以上は特定の知識ベースを例にして示したものだが、リフォメーションの効果は概ねこのような傾向を示す。

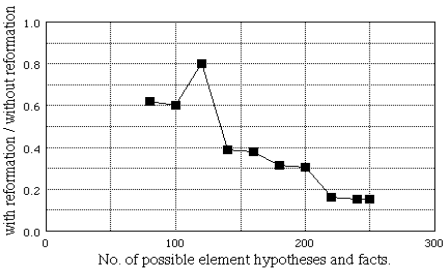

図 5　リフォメーションによる生成命題ルール数の減少効果（別の知識ベースのとき）

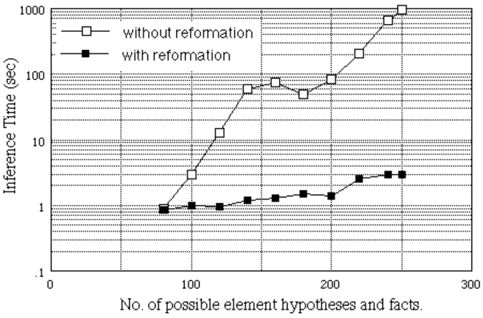

図 6　リフォメーションによる推論時間の高速化効果（別の知識ベースのとき）

図 5, 図 6 は各種タイプの述語ホーン節知識が含まれているコストに基づく仮説推論問題に対する同じ実験結果を示している。変動は見られるが、問題サイズの増大とともに本論文の述語ルール・リフォメーションによる推論時間の減少効果は顕著になってきている。

5. むすび

命題レベルの推論問題には高速解法が考案、開発されているが、表現力が高い述語論理レベルとなると、なかなか良い高速解法を見い出すことが難しい状況にある。本論文ではコストに基づく述語論理版仮説推論の解法として、述語ルールを命題ルールに展開し、高速命題版仮説推論法を適用して準最適解を求める効率的な手法を示した。特に、演繹データベースの手法である QSQR 法により与えられたゴールの証明に関係する部分のみを抽出して命題ルールに展開するに際して、この展開を効率的に行えるようにし、生成される命題ルール数の削減を図る述語ルールのリフォメーションが、本手法を効果的なものにすることを明らかにした。今後の主な課題は以下の項目である。

- 本論文に記した述語ルールの 3 種のリフォメーションは範囲が限定されたものであり、また効果の定

量的評価もまだ十分でない。より広範囲な表現に対して効果的な方法を見い出すと共に、効果をより明確に評価できるようにする必要がある。

- 与えられたゴールに関係する述語ルールを抽出して命題ルールに展開するのに、ここではQSQR法を利用した。このQSQR法は演繹データベースの効率的問い合わせ処理の手法であり、ここでの述語ルールの命題ルールへの展開という目的のためにはより適した手法があるはずで、探求する必要がある。
- 各種処理モジュールを統合し、統合されたシステムとして利用可能にする。

謝　辞

議論いただいたDr. Helmut Prendinger (Univ. of Salzburg/東大)、及び記述の改善のコメントをいただいた査読者に感謝します。

◇ 参考文献 ◇

[Bylander89] T.Byladner et al.: Some Results Concerning the Complexity of Abduction, Proc.1st Conf. on Principles of Knowledge Representation and Reasoning (KR'89) (1989)

[Charniak90] E.Charniak and S.Shimony: Probabilistic Semantics for Cost Based Abduction, Proc.AAAI-90, pp.106-111 (1990)

[堂前 94] 堂前、石塚：仮説推論高速化のための知識ベースリフォメーション、人工知能学会誌、Vol.9, No.4, pp.595-603 (1994)

[福田 97] 福田、石塚：ネットワーク化バブル伝播法による述語版仮説推論システムの構築、情処全大、No.5G-8 (1997.3)

[古川 87] 古川、溝口(共編)：プログラム変換、共立出版 (1987)

[Gu93] J.Gu: Local Search for Satisfiability(SAT) Problem, IEEE Trans. SMC, Vol.23, No.4, pp.1108-1129 (1993)

[石塚 94] 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)

[石塚 96] 石塚満：知識の表現と高速推論 – 第6章：仮説推論とその高速化メカニズム、丸善 (1996)

[Kautz89] H.Kautz and B.Selman: Hard Problems for Simple Default Logics, Proc.1st on Principles of Knowledge Representation and Reasoning (KR'89) (1989)

[近藤 93] 近藤、牧野、石塚：述語論理知識を扱う高速仮説推論システム、人工知能学会誌、Vol.8, No.6, pp.819-827 (1993)

[Kondo93] A.Kondo, T.Makino and M.Ishizuka: Efficient Hypothetical Reasoning System for Predicate-logic Knowledge Base, Knowledge-Based Systems, Vol.6, No.2, pp.87-94 (1993)

[近藤 94] 近藤、石塚 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.290-298 (1994)

[Kondo96] A.Kondo and M.Ishizuka: Efficient Inference Method for Computing an Optimal Solution in Predicate-Logic Hypothetical Reasoning, Knowledge-Based Systems, Vol.9, pp.163-171 (1996)

[松尾 98] 松尾、二田、石塚：SL法：線形・非線形計画法の併用によるコストに基づく仮説推論の準最適解計算、人工知能学会誌、Vol.13, No.6, pp.953-961 (1998.11)

[西尾 88] 西尾、楠見 演繹データベースにおける再帰的な問い合わせ評価法、情報処理、Vol.29, No.2, pp.240-255 (1998)

[大澤 94] 大澤、石塚：仮説推論における準最適解を多項式時間で計算するネットワーク化バブル伝播法、電子情報通信学会誌、Vol.J77-D-II, No.9, pp.1817-1829 (1994)

[大澤 95a] 大澤、石塚：改良型ネットワーク化バブル伝播法による低次多項式時間仮説推論法、人工知能学会誌、Vol.10, No.5, pp.731-740 (1995)

[大澤 95b] 大澤、石塚：多項式時間仮説推論を達成するネットワーク化バブル伝播法の述語論理への拡張、人工知能学会誌、Vol.11, No.5, pp.731-740 (1995)

[Ohsawa97] Y.Ohsawa and M.Ishizuka：Network Bubble Propagation: A Polynomial-time Hypothetical Reasoning Method for Comuting Near-optmal Solutions, Artif.Intelli., Vol.91, No.1, pp.131-154 (1997)

[岡本 93] 岡本、石塚：整数計画法の近似解法を適用した準最適解計算の高速仮説推論法、人工知能学会誌、Vol.8, No.2, pp.222-229 (1993)

[Santos94] E.Santos Jr.: A Linear Constraint Satisfaction Approach to Cost-based Abduction, Artificial Intelligence, Vol.65, pp.1-27 (1994)

[Selman93a] B.Selman and H.Kautz: An Empirical Study of Greedy Local Search for Satisfiability Testing, Proc.AAAI-93, pp.46-51 (1993)

[Selman93b] B.Selman and H.Kautz: Domain-independent Extension to GSAT: Solving Large Structured Satisfiability Problems, Proc.IJCAI-93, pp.290-295 (1993)

[Selman97] B.Selman, H.Kautz and D.McAllester: Ten Challenges in Propositional Reasoning and Search, Proc.IJCAI-97, pp.50-54 (1997)

[Vielle86] L.Vieille: Recursive Axious in Deductive Databases: The Query/Subsequery Approach, Proc.1st Int'l Conf. on Expert Database Systems, pp.179-193 (1986)

〔担当編集委員：××○○，査読者：○○××〕

著者紹介

棚橋　一也（非会員）

1998年 東京大学工学部電子情報工学科卒業。在学中に知識リフォメーションによる推論の高速化の研究を行った。現在、(株)富士通に勤務し、金融システムの開発に従事。

⟨kazuya@happy.email.ne.jp⟩

福田　茂紀（非会員）

1997年 東京大学工学部電子情報工学科卒業、1999年、同大学院修士課程修了。在学中に述語版仮説推論システムと並列論理プログラミング言語の研究を行った。現在、(株)富士通研究所、ネットメディア研究センターに所属。

⟨fukuta@flab.fujitsu.ne.jp⟩

石塚　満（正会員）

は前掲(Vol.14, No.1, p.101)参照。

⟨shizuka@miv.t.u-tokyo.ac.jp⟩