論理和データベースに対するゴール指向問合せ処理

島 尻 優 香† 世 木 博 久† 伊 藤 英 則†

本論文では，論理和データベースに対するゴール指向問合せ処理方法を提案する．節集合の充足可能性を調べるアルゴリズムとして，「関連性」の概念を導入したSATCHMORE[9]が提案されている．しかし，問合せ処理では，真，偽，不定を区別したい場合があるため，「関連性」の概念をそのまま用いても十分ではない．本論文では，まず問合せ処理のために変更を加えた関連性の概念に基づいた問合せ処理方法を提案する．そして，この手法が，不確定論理データベースのあるクラスに対して，健全かつ完全であることを示す．

Goal-directed Query Processing in Disjunctive Logic Databases

YUKA SHIMAJIRI,† HIROHISA SEKI† and HIDENORI ITOH†

We propose a goal-directed query processing method for disjunctive logic databases. The notion of "relevancy" proposed in a theorem prover called SATCHMORE in 9) is useful for the satisfiability checking of a given set of clauses, but it is not sufficient for the query processing in disjunctive databases, where it is essential in some applications for the answer to a given query to be distinguished between "true", "false", or "unknown (possibly true)". We give a goal-directed query processing procedure based on the relevancy, but suitably modified for the above purpose, and show that our procedure is sound and complete for a class of positive disjunctive databases in the minimal model semantics.

1. はじめに

論理和データベース（disjunctive databases）については，今までに多くの研究結果が報告されている（たとえば文献10)）．論理和データベースは，知識表現の枠組みの1つであり，また，数学などの分野における定理証明やアブダクション[21]のような応用が期待されている．ただし，論理和データベースについて，意味論はすでに数多く提案されているが，それに対する効率的な問合せ処理手続きについては，研究の余地があると考えられる[1]．

確定節データベースに対しては，トップダウン型・ボトムアップ型の両方の問合せアルゴリズムが数多く知られている（たとえば文献2)）．それらのアルゴリズムでは，主に繰り返し現れるサブゴールの同じ証明を省くことや，与えられたゴールを証明するのに必要なサブゴールのみに注目するといったゴール指向性によって，効率の良さが実現されている．しかしながら，論理和データベースの場合は，節の頭部に現れる選言がそれらの効率化を難しいものにしている．たとえば，SATCHMO[11]や MGTP（Model Generation Theorem Prover）[12]のようなボトムアップ計算は，そのメカニズムは単純であるが，問合せ処理におけるゴール指向性が失われるため，与えられたゴールの証明とは無関係な計算も行ってしまう．

本論文では，論理和データベースに対するゴール指向問合せ処理手続きを提案する☆．そのために，定理証明系 SATCHMORE[9]で導入されている「関連性」の概念を用いる．関連性の概念は，効率良く推論を行うための節を選ぶときに基準として用いられており，与えられた節集合の充足可能性を判定するためには有効である．しかるに，最小モデル意味論では一般的に複数のモデルが存在する．そのため，問合せに対する答として，それが成り立つと確定するもの（真），成り立たないと確定するもの（偽），どちらにも確定しないもの（不定）が存在し，それらを区別したい場合がある．その区別をするためには，SATCHMORE で用いられている「関連性」の概念では十分ではない．そこで，本論文では，問合せ処理のために変更を加えた

† 名古屋工業大学
Nagoya Institute of Technology

☆ 本論文は，文献22)の報告をもとに論文にまとめたものである．

関連性の概念に基づいた問合せ処理方法を提案する．

以下に論文の構成を示す．2 章では，本研究における問合せ処理の枠組みと従来の研究について述べる．3 章では,「関連性」の定義と「関連性」の概念に基づく問合せ処理手続きについて説明し，その正当性を示す．4 章では，実験結果を示す．

2. 論理和データベースに対する問合せ処理

本論文で扱う論理和データベースは，以下のような節の有限集合であるとする．

$$A_1 \vee \ldots \vee A_l \leftarrow B_1 \wedge \ldots \wedge B_m \ (l,\ m \geq 0) \tag{1}$$

ただし，$A_i, B_j (0 \leq i \leq l, 0 \leq j \leq m)$ はアトムとする．

本論文では，各節は関数記号を含まず，領域限定 (range-restricted)☆であるとする．

論理和データベースの意味は，最小モデル意味論[13]により与えられる．$\mathcal{MM}_P$ を節集合 P の最小モデルの集合とし，基底の問合せ Q が与えられたとする．このとき，Q に対する答（真，偽，不定）は，以下のように定義される．

Q が真である． iff $\forall M_i \in \mathcal{MM}_P\ M_i \models Q$

Q が偽である． iff $\forall M_i \in \mathcal{MM}_P\ M_i \not\models Q$

Q が不定である． iff Q は真でも偽でもない．

以下では，与えられる問合せQ は基底であると仮定する．

2.1 関連研究

これまでに，論理和データベースに対する証明手続きは多くの提案がなされている（文献 4), 5), 10), 14), 17), 23) など)．

たとえば，トップダウン型の線形導出法の 1 つである SLI 導出は，ゴール指向性を持っている．しかしながら，先祖導出（ancestry resolution）や簡約化 (factoring) のように幾分複雑な操作を含んでおり，SLD 導出のように単純な手続きではない．

一方，ボトムアップ型の証明手続きとしては，MGTP[12]がある．Inoue らは，この MGTP を基にした論理和データベースに対する証明手続きを提案している[6]．

MGTP は，初期モデル候補 $I_0 = \phi$ から出発して，以下の操作を繰り返すことにより，アトムの集合であるモデル候補 I を生成する．

(i) **モデル拡張操作：** 以下のような節が存在し，

$$A_1 \vee \cdots \vee A_l \leftarrow A_{l+1} \wedge \cdots \wedge A_m \ (l \geq 1)$$

$A_{l+1}\sigma, \ldots, A_m\sigma \in I$ かつ，$\forall i = 1, \ldots, l$ $A_i\sigma \notin I$ が成り立つとき，各 $i = 1, \ldots, l$ に対し，$I \cup \{A_i\sigma\}$ なる l 個のモデル候補を生成する．

(ii) **モデル棄却操作：** 以下のような負節が存在し，

$$\leftarrow A_1, \ldots, A_m$$

$A_1\sigma, \ldots, A_m\sigma \in I$ を満たす代入 σ が存在するとき，I を棄却する．

このように，この MGTP は非常に単純なメカニズムであるが，次の例に示すとおりゴール指向性はない．

例 2.1 以下のデータベースに問合せ q が与えられたとする．

$$q \leftarrow r \tag{c1}$$
$$a \vee b \leftarrow \tag{c2}$$
$$p \vee c \vee d \leftarrow \tag{c3}$$
$$p \vee r \leftarrow \tag{c4}$$
$$\leftarrow p \tag{c5}$$

MGTP により推論を行うと，以下のようなモデル候補の集合 S_i が得られる．

$S_0 = \{\phi\}$

$S_1 = \{\{a\}, \{b\}\}$

$S_2 = \{\{a,p\}, \{a,c\}, \{a,d\}, \{b,p\}, \{b,c\}, \{b,d\}\}$

節 (c2) および節 (c3) は，明らかに問合せ q を証明するのには必要ない．しかしながら，それらの節が用いられることにより，上記のような，問合せを証明するのには貢献しないアトムを含むモデル候補が数多く生成されてしまう．ここで，ϕ を節 (c4)，節 (c1) を用いて拡張したとすると，

$S_{1'} = \{\{p\}, \{r\}\}$

$S_{2'} = \{\{r, q\}\}$

が得られ，前者のような多数のモデル候補を生成しないで q を証明することができる． □

このように，ボトムアップ型の証明手続きの探索空間の大きさは，証明に用いる非ホーン節に依存する．

2.2 混合型の問合せ処理

最近 Loveland らによって，非ホーン節の選択を制御するボトムアップ型の充足可能性判定アルゴリズム SATCHMORE (SATCHMO with RElevancy)[9]が提案されている．この手続きは，トップダウン，ボトムアップの両方を用いる混合型の (hybrid) 計算により，与えられた節集合の充足可能性を判定する．すなわち，節集合 P をホーン節集合 P_H と非ホーン節集合 P_{nH} に分け，P_H については，トップダウンである SLD 導出 (Prolog) を用い，P_{nH} については，ボトムアップの前向き計算を行う．

☆ 節が領域限定であるとは，節の頭部に現れる変数はすべて本体にも現れているということである．

P_{nH} に対するボトムアップ計算の制御には，関連性（以下に定義を示す）の概念を用いる．それによって，関連のない非ホーン節を適用したときに起こる探索空間の爆発を防ぐことができる．ただし，最小モデル意味論では一般的に複数のモデルが存在する．そのため，問合せ処理では，真，偽，不定を区別したい場合がある．その区別をするためには，SATCHMORE で用いられている「関連性」の概念では十分ではない．

そこで，本論文では，真，偽，不定を区別する処理を，真か否かを区別する処理と偽か不定を区別する処理に分け，そのそれぞれの目的に合わせて変更を加えた関連性を用いて節の選択の制御を行う．以下では，変更を加えた関連性が，論理和データベースに対する問合せ処理において有効に働くことを示す．

3. 関連性に基づくゴール指向問合せ処理

3.1 関連性

以下に，関連性[9]に関する定義を示す．

定義 3.1（I-失敗アトム） P_H をホーン節集合，I を基底アトムの集合とする．アトム B が P_H に関して I-失敗アトムであるとは，B が $P_H \cup I$ の論理的帰結でないことである（つまり，$P_H \cup I \not\vdash B$）． □

P をホーン節集合，$\leftarrow G$ をゴール（G はアトムの連言），また，$P \cup \{\leftarrow G\}$ の SLD 反駁木☆は成功葉を持たないとする．このとき，その有限失敗枝において選択されたアトムすべての集合を $\mathcal{FB}_P(\leftarrow G)$ と表す．

定義 3.2（全関連性） P_H，P_{nH} をそれぞれホーン節集合，非ホーン節集合，P_{neg} を負節の集合，I を基底アトムの集合とする．このとき I-失敗アトム B が $(P_{neg}, P_H \cup I)$ において**全関連アトム**であるとは，以下の条件のいずれかを満たすことをいう．

(i) $B \in \mathcal{FB}_{P_H \cup I}(\leftarrow \Gamma)$．ただし，$\leftarrow \Gamma$ は P_{neg} 中の負節とする．あるいは，

(ii) $B \in \mathcal{FB}_{P_H \cup I}(\leftarrow B_1 \wedge \cdots \wedge B_m)$．ただし，以下のような非ホーン節 $C \in P_{nH}$
$$A_1 \vee \ldots \vee A_l \leftarrow B_1 \wedge \ldots \wedge B_m \ (l \geq 2, m \geq 0)$$
が存在し，すべての A_i $(l \geq i \geq 1)$ が $(P_{neg}, P_H \cup I)$ において全関連であるアトムと共通のインスタンスを持っているとする．□

定義 3.3（部分関連性） P_H，P_{nH}，P_{neg}，I は定義 3.2 と同様とする．このとき，I-失敗アトム B が，$(P_{neg}, P_H \cup I)$ において**部分関連アトム**であるとは，以下の条件のいずれかを満たすことをいう．

(i) $B \in \mathcal{FB}_{P_H \cup I}(\leftarrow \Gamma)$．ただし，$\leftarrow \Gamma$ は P_{neg} 中の負節とする．あるいは，

(ii) $B \in \mathcal{FB}_{P_H \cup I}(\leftarrow B_1 \wedge \cdots \wedge B_m)$．ただし，以下のような非ホーン節 $C \in P_{nH}$
$$A_1 \vee \ldots \vee A_l \leftarrow B_1 \wedge \ldots \wedge B_m \ (l \geq 2, m \geq 0)$$
が存在し，**少なくとも 1 つの** A_i が $(P_{neg}, P_H \cup I)$ において部分関連であるアトムと共通のインスタンスを持っているとする． □

直観的には，あるアトム B が $(P_{neg}, P_H \cup I)$ において全（部分）関連であるとは，$P_H \cup I$ において，B が矛盾を導くのに貢献する可能性があるということを意味する．

また，非ホーン節が $(P_{neg}, P_H \cup I)$ において全（部分）関連節であるとは，その頭部のアトムがすべて（少なくとも 1 つ）全（部分）関連アトムと共通のインスタンスを持っていることをいう．

非ホーン節 $C = \Sigma \leftarrow \Gamma$ が $P_H \cup I$ に対して，$P_H \cup I \vdash \Gamma\theta$ を満たすある基底化代入（ground substitution）θ が存在し，かつ $P_H \cup I \not\vdash \Sigma\theta$ が成り立つとき，$C\theta$ を違反節[11]という．

ここで以下の表記を導入する．

$TRV(P_{neg}, I)$
$=\{C\theta \mid C$ は (P_{neg}, I) において全関連節であり，かつ $C\theta$ が違反節である.$\}$

$PRV(P_{neg}, I)$
$=\{C\theta \mid C$ は (P_{neg}, I) において部分関連節であり，かつ $C\theta$ が違反節である.$\}$

例 3.1 例 2.1 と同じデータベースについて考える．$I = \phi$ とすると，$\mathcal{FB}_{P_H \cup I}(\leftarrow q) = \{q, r\}$ であり，q と r が $(\{\leftarrow q\}, P_H \cup I)$ に関して部分関連アトムである．よって，$TRV(\{\leftarrow q\}, I) = \phi$，$PRV(\{\leftarrow q\}, I) = \{(c4)\} = \{p \vee r \leftarrow\}$ となる． □

3.2 関連性に基づくゴール指向問合せ処理

この節では，本論文で紹介する問合せ処理手続き（以下，TPR と呼ぶ）について説明する．その流れを**図 1** に示す．TPR には，入力として基底の問合せ Q と節集合 $P = P_H \cup P_{nH}$ が与えられる．本論文では，P は充足可能であると仮定する．

TPR は，基本的に次の 2 つの処理部からなっている．第一の処理部（4 行目から 15 行目）を TR 処理部と呼び，ここでは Q が真か否かを判定する．第二処理部（16 行目から 31 行目）を PR 処理部と呼び，ここでは Q が不定か否かを判定する．

図 1 において，I_j は基底アトムの集合であり，P のモデル候補と呼ぶ．I_j は手続き中で MGTP のよう

☆ 以下では，SLD 反駁におけるゴール中のアトムの選択の規則（computation rule）は，1 つ与えられており，固定されているとする．

```
procedure TPR
入力 : a query Q, a program P = P_H ∪ P_nH (P_neg ⊆ P_H)
出力 : true, false, or unknown
1  begin
2    S_0 := {φ} ;
3    i := 0;
   % TR 処理部
4    while ∀I_j ∈ S_i (P_H ∪ I_j ⊨ Q or
                 TRV(P_neg ∪ {← Q}, I_j) ≠ φ ) do
5      if ∀I_j ∈ S_i P_H ∪ I_j ⊨ Q then
6        return true % Q は true
7      else
8        foreach I_j ∈ S_i do
9          if TRV(P_neg ∪ {← Q}, I_j) ≠ φ then
10           begin
11             choose a clause C from
                        TRV(P_neg ∪ {← Q}, I_j) ;
               % C = A_1 ∨ ··· ∨ A_m ← B_1 ∧ ··· ∧ B_n
12             S_t := {{A_k} ∪ I_j | 1 ≤ k ≤ m,
                        P_H ∪ I_j ∪ {A_k} : sat.};
13             S_{i+1} := S_i ∪ S_t − {I_j} ;
14             i := i + 1
15           end ;
   % PR 処理部
16   while ∃I_j ∈ S_i
     ( P_H ∪ I_j ⊨ Q or PRV({← Q}, I_j) ≠ φ) do
17     if ∃I_j ∈ S_i P_H ∪ I_j ⊨ Q then
18       begin
19         choose I_j s.t P_H ∪ I_j ⊨ Q;
20         if MIN(I_j, Q) ≠ φ then
21           return unknown % Q は unknown
22         else S_i := S_i − {I_j}
23       end
24     else
25       begin
26         choose I_j s.t. PRV({← Q}, I_j) ≠ φ ;
27         choose a clause C from PRV({← Q}, I_j) ;
28         S_t := {{A_k} ∪ I_j | 1 ≤ k ≤ m,
                        P_H ∪ I_j ∪ {A_k} : sat.};
29         S_{i+1} := S_i ∪ S_t − {I_j} ;
30         i := i + 1
31       end ;
32   return false % Q は false
33 end
```

図1 問合せ手続き

Fig. 1 Query processing procedure.

に非ホーン節を用いて拡張される．S_i $(i \geq 0)$ は，I_j $(1 \leq j \leq |S_i|)$ の集合であるとする．

まず，初期集合 $S_0 = \{\phi\}$ とする（2 行目）．

TR 処理部では，S_i の要素である各 I_j において $P_H \cup I_j \models Q$ または $TRV(P_{neg} \cup \{\leftarrow Q\}, I_j) \neq \phi$ である間（4 行目），以下の操作を繰り返す．

- すべての I_j に対して，$P_H \cup I_j \models Q$ が満たされるかどうかを調べる．
- もし満たされているならば，補題 A.2.4 より，与えられた問合せに対する答は，真となる．
- そうでないとき，この手続きは，各 I_j に対して，$TRV(P_{neg} \cup \{\leftarrow Q\}, I_j)$ 中の節を用いてモデル候補を拡張する（12 行目）．

4 行目の条件が満たされないとき，TR 処理部を抜け，PR 処理部に移る．

このとき，補題 A.2.1 より，Q を真としない P のモデルが存在することが分かる（つまり，$P \cup \{\leftarrow Q\}$ は充足可能）．PR 処理部では，手続きは Q を真とする P のモデルが存在するか否かのみを調べる．そこで，$P_H \cup I_j \models Q$ または，$PRV(\{\leftarrow Q\}, I_j) \neq \phi$ を満たす $I_j \in S_i$ が存在する間以下の操作を繰り返す（16 行目）．

- S_i に関して
$$\exists I_j \in S_i \quad P_H \cup I_j \models Q \tag{2}$$
が満たされるか否か調べる．
- (2) が満たされるならば（17 行目），以下の関数（付録 A.1 にそのアルゴリズムを示す）を用いて Q を真とする P の最小モデルが存在するか否かを調べる．
$$MIN(I_j, Q) = \{M \mid \exists M \in MG^*_P(I_j), M \in \mathcal{MM}_P \text{かつ} M \models Q\} \tag{3}$$
ただし，$MG^*_P(I_j)$ は I_j を初期集合として MGTP により生成されたモデル集合とする．
- ここで，$MIN(I_j, Q) \neq \phi$ であれば，その問合せに対する答は不定である．
- (2) の条件が満たされないとき，$PRV(\{\leftarrow Q\}, I_j)$ 中の節を選びモデル候補を拡張する（27 行目から 29 行目）．

16 行目の条件が満たされないとき，補題 A.2.3 より，与えられた問合せが偽であると判断する（32 行目）．

例 3.2 以下のデータベース P^{ex1} について考える．

Horn clauses P^{ex1}_H:

$$\begin{array}{rcll} a_1 & \leftarrow & c_1 & \text{(hc1)} \\ b_1 & \leftarrow & & \text{(hc2)} \\ d_1 & \leftarrow & c_1 \wedge c_2 & \text{(hc3)} \\ e & \leftarrow & a_1 & \text{(hc4)} \\ e & \leftarrow & c_2 & \text{(hc5)} \\ & \leftarrow & a_2 & \text{(hc6)} \end{array}$$

non-Horn clauses $P^{ex1}_{>1}$:

$$\begin{array}{rcll} a_1 \vee a_2 & \leftarrow & b_1 \wedge b_2 & \text{(hc7)} \\ b_2 \vee b_3 & \leftarrow & & \text{(hc8)} \\ c_1 \vee c_2 & \leftarrow & b_1 & \text{(hc9)} \\ d_2 \vee d_3 \vee d_4 & \leftarrow & & \text{(hc10)} \end{array}$$

（場合 1） 問合せ e が与えられたとする．まず，$S_0 = \{\phi\}$ とする．このとき，$I^0_1 = \phi$ であり
$$TRV(\{(\text{hc6}), \leftarrow e\}, I^0_1) = \{(\text{hc9})\}$$
となる．$P_H \cup I^0_1 \not\models e$ であるので，(hc9) が選ばれる (11 行目)．$P_H \cup \{c_1\}$ および $P_H \cup \{c_2\}$ が無矛盾であるので，$S_1 = \{\{c_1\},\{c_2\}\}$ が生

成される．このとき，$P_H \cup \{c_1\} \models e$ および $P_H \cup \{c_2\} \models e$ より，$TPR(e,P^{ex1}) = \mathrm{true}$ となる．

（場合 2）次に，問合せ a_1 が与えられたとする．TR 処理部では，$I_1^0 = \phi$ に関して全関連で違反である節は存在しない．そこで，PR 処理部に移る．ここで，$PRV(\{\leftarrow a_1\}, I_1^0) = \{(\mathrm{hc8}), (\mathrm{hc9})\}$ から，節 (hc8) が選ばれたとすると (27行目)，$S_1 = \{\{b_2\},\{b_3\}\}$ が生成される．$P_H \cup I_j^1 \models Q$ なる $I_j^1 \in S_1$ が存在しないので，$PRV(\{\leftarrow a_1\},\{b_2\}) = \{(\mathrm{hc7})\}$ から節 (hc7) が選ばれ，$S_2 = \{\{a_1,b_2\},\{b_3\}\}$ が生成される．ここで，$I_1^2 = \{a_1, b_2\}$ とすると

$$MIN(I_1^2, a_1) = \{\{a_1, b_1, b_2, e, d_2\}, \cdots\}$$

より，$TPR(a_1,P^{ex1}) = \mathrm{unknown}$ となる．

（場合 3）次に，問合せ d_1 が与えられたとする．TR 処理部では，$I_1^0 = \phi$ に関して全関連で違反である節は存在しないので，PR 処理部に移る．$PRV(\{\leftarrow d_1\},\{\phi\}) = \{(\mathrm{hc9})\}$ から，節 (hc9) が選ばれ，$S_1 = \{\{c_1\},\{c_2\}\}$ が生成される．ここで，$PRV(\{\leftarrow d_1\},\{c_1\}) = \phi$ と $PRV(\{\leftarrow d_1\},\{c_2\}) = \phi$ より，$TPR(d_1,P^{ex1}) = \mathrm{false}$ となる．

この例において，生成された集合 S_i の最終的な大きさは，それぞれのケースにおいて 2 であった．一方，関連性についての解析を行わず，単純に違反節を用いてモデル候補を拡張した場合，最終的な S_i の大きさは，(最悪で) 12 となる．この例から，論理和データベースに対する問合せ処理において関連性の概念を利用することは，効果的であることが分かる． □

本論文で提案する手続きにおいて，関連性の概念は主に以下のような 2 つの役割がある．まず，推論に用いる非ホーン節を選択するための規準を与えていることである．これにより，問合せや矛盾を導く可能性のない非ホーン節を選ばなくなり，不必要な枝分かれを抑えることができる．もう 1 つは，全関連節が存在しないことより Q が真ではないことが，また部分関連節が存在しないことより Q が偽であることが判別できる．全関連と部分関連をこの順序で用いることによって，より効率的な問合せに対する答の判定を可能にしている．

以下の定理により，手続き TPR の妥当性を示す (証明は，付録 A.2 に示す)．

定理 3.1（アルゴリズム $\boldsymbol{TPR}$ の正当性） P を定数記号以外の関数記号を含まず，領域限定である論理和データベースとする．ここで，基底の問合せ Q が与えられたとき，

Q が真 iff $TPR(P,Q) = \mathrm{true}$
Q が不定 iff $TPR(P,Q) = \mathrm{unknown}$
Q が偽 iff $TPR(P,Q) = \mathrm{false}$

が成り立つ． □

手続き TPR には次の 2 つの拡張が考えられる．

(i) 本論文では，定数記号以外の関数記号を含まないデータベースのみを対象とした．これは，アルゴリズムの停止条件のためである．与えられたデータベースに関数記号が含まれていた場合，手続きは健全であるが，完全性を失う．手続きの停止性を保証するための統語的条件は，今後の研究課題であると考えられる．

(ii) 本論文では与えられる問合せは基底であると仮定している．けれども，たとえば $Q(X)$ のように変数を含む問合せが与えられたとすると，次のようにして解代入を計算することができる．与えられたデータベース P に新しい節 $Q \leftarrow Q(X)$ を加え，$(Q, P \cup \{Q \leftarrow Q(X)\})$ を入力として TPR を実行する．ここで，問合せ Q が真または不定の場合，TPR によって，最終的に生成された各アトム集合 $I_j \in S_n$ $(1 \le j \le k; k > 0)$ の中の $Q(X)$ の代入例を $Q(t_j) \in I_j$ とする．このとき，$\{t_1/X, \ldots, t_k/X\}$ を，問合せ $Q(X)$ の解代入として得ることができる．

4. 実験結果

本論文で提案した手続きの有効性を確かめるために，試作プログラムで実験を行った．プログラムは，SICStus Prolog により実装し，SS-5 上で実行時間を測定した．**表 1** に，以下のデータベースを用いて実験を行った結果を示す．

Horn clauses : P_H^{ex3}

$$\begin{aligned}
has(a, gym) &\leftarrow \\
has(b, gym) &\leftarrow \\
has(c, gym) &\leftarrow \\
has(d, pool) &\leftarrow \\
has(e, pool) &\leftarrow \\
wet(X) &\leftarrow swims(X) \\
wet(X) &\leftarrow rain \\
&\leftarrow rain \wedge fine
\end{aligned}$$

non-Horn clauses : P_{nH}^{ex3}

$$\begin{aligned}
fine \vee cloudy \vee rain &\leftarrow \\
trains(X) \vee plays(X, pp) &\leftarrow has(X, gym) \\
swims(X) \vee rain &\leftarrow has(X, pool)
\end{aligned}$$

表 1 は，手続き TPR，単純なボトムアップ計算方

表 1 実行時間の比較
Table 1 Comparison of the runtimes.

問合せ	TPR	PR^2	$withoutR$	$\frac{TPR}{without\ R}$
query1	0.017	0.043	0.506	0.034
query2	0.073	0.180	1.769	0.041
query3	0.130	0.140	0.809	0.161

query1 : $wet(e)$ (true)
query2 : $swims(e)$ (unknown)
query3 : $swims(a)$ (false)

法（*without R* で示す）および，部分関連だけを用いた計算方法（PR^2 で示す）の比較を示した．表より，今回用いた例題では TPR の実行時間は，単純なボトムアップ計算と比べて 16.1 % 以下であることが分かる．

5. おわりに

本論文では，論理和データベースに対するゴール指向問合せ処理を提案した．まず，全関連性，および部分関連性を用いた手続き TPR を示した．それらの関連性に基づく制御は，最小モデル意味論に基づく問合せ処理において有効に働いている．

本論文では，否定を含まない論理和プログラムを対象としたが，Inoue らが提案しているボトムアップ計算手続き[6]に関連性の概念を加えることによって，本手続きは，失敗による否定も含む論理和プログラムに拡張できるものと考えている．

また，手続き TPR に基づく問合せ処理システムを試作し，いくつかの例題で実験を行った．それにより，与えられた問題に依存するものの，実行時間が約16%以下に縮小されることを確認した．

さらに，静的なプログラム解析（たとえば文献7)）を用いた拡張によって，探索空間を絞り込むことなども考えられる．今後の課題の1つとして，並列計算機上の実装などが考えられる．

参考文献

1) Baral, C. and Gelfond, M.: Logic Programming and Knowledge Representation, *J. of Logic Programming*, Vol.19/20, pp.73–148 (1994).
2) Bancilhon, F. and Ramakrishnan. R.: An Amateur's Introduction to Recursive Query Processing Strategies, *Proc. ACM SIGMOD Intl. Conf. on Management of Data*, pp.16–52 (1986).
3) Fujita, M., Slaney, J. and Bennet, F.: Automatic Generation of Some Results in Finite Algebra, *Proc. IJCAI93*, pp.52–57 (1993).
4) Grant, J. and Minker, J.: Answering Queries in Indefinite Database and the Null Value Problem, *Advance in Computing Theory* 3, *The Theory of Databases*, Kanellakis, P. (Ed.), pp.247–267, JAI Press, Greenwich (1986).
5) Hensen, L. and Prak, H.: Compiling the GCWA in Indefinite Deductive Database, *Foundations of Deductive Database and Logic Programming*, Morgen Kaufmann (1988).
6) Inoue, K., Koshimura, M. and Hasegawa, R.: Embedding Negation as Failure into a Model Generation Theorem Prover, *Proc. 11th Int'l Conf. on Automated Deduction* (1992).
7) Kato, S., Seki, H. and Itoh, H.: An Efficient Abductive Reasoning System Based on Program Analysis, *Proc. the 3rd Int'l WSA*, LNCS, Vol.724, pp.230–241 (1993).
8) Kowalski, R.: Logic Programming in Artificial Intelligence, *Proc. IJCAI'91*: pp.596–603 (1991).
9) Loveland, D.W., Reed, D.D. and Willson, D.S.: SATCHMORE: SATCHMO with RElevancy, *J. of Automated Reasoning*, Vol.14, pp.325–351 (1995).
10) Lobo, J., Minker, J. and Rajaseker, A.: *Foundations of Disjunctive Logic Programming*, MIT Press (1992).
11) Manthey, R. and Bry, F.: SATCHMO: A Theorem Prover Implemented in Prolog, *Proc. CADE'88* (1987).
12) Fujita, H. and Hasegawa, R.: A Model Generation Theorem Prover in KL1 Using Ramified Stack Algorithm, *Proc. ICLP'91*, pp.535–548 (1991).
13) Minker, J.: On Indefinite Data Bases and the Closed World Assumption, *Proc. 6th Int'l Conf. on Automated Deduction*, LNCS, Vol.138, pp.292–308, Springer-Verlag (1982).
14) Minker, J. and Rajasekar, A.: A Fixpoint Semantics for Disjunctive Logic Programs, *J. of Logic Programming*, Vol.9, No.1, pp.45–74 (1990).
15) Niemelä, I.: A Tableau Calculus for Minimal Model Reasoning, *Proc. the Fifth Workshop on Theorem Proving with Analytic Tableaux and Related Methods* (Terrasini, Italy, May 15-17, 1996), LNAI, Vol.1071, pp.143–159, Springer-Verlag (1996).
16) Poole, D.: A Logical Framework for Default Reasoning, *Artificial Intelligence*, Vol.36, pp27–47 (1988).
17) Ross, K.A. and Topor, R.W.: Inferring Negative Information from Disjunctive Databases, *Journal of Automated Reasoning*, Vol.4, pp.397–424 (1988).

18) Rajaseker, A., Lobo, J. and Minker, J.: Weak Generalized Closed World Assumption, *Journal of Automated Reasoning*, Vol.5, pp.293–307 (1990).
19) Ramsay, A.: Generating Relevant Models, *Journal of Automated Reasoning* Vol.7, pp.359–368 (1991).
20) Sakama, C. and Inoue, K.: Negation in Disjunctive Logic Programs, *Proc. ICLP'93*, pp.703–719 (1993).
21) Sakama, C. and Inoue, K.: On the Equivalence between Disjunctive and Abductive Logic Programs, *Proc. ICLP'94* (1994).
22) Shimajiri, Y., Seki, H. and Itoh, H.: Goal-Directed Query Processing in Disjunctive Logic Databases, *Proc. PLILP'95*, LNCS, Vol.982, pp.415–430, Springer-Verlag (1995).
23) Yahya, A. and Henschen, L.J.: Deduction in Non-horn Databases, *Journal of Automated Reasoning*, Vol.1, No.2, pp.141–160 (1993).

付　　録

A.1 $MIN(I,Q)$ のアルゴリズム

$MIN(I,Q)$ は，PR 処理部において Q を真とするモデル候補 I が生成されたとき，答が不定であることを確かめるために，I を部分集合として持つ，論理和データベース P の最小モデルを生成する手続きである．ここで以下の表記を導入する．

$Ground(P)=\{C\theta \mid C\in P$ かつ θ は基底化代入．$\}$

この手続きでは，最小モデルの次の性質を用いている[15]．

命題 A.1.1 P を定数記号以外の関数記号を含まず，領域限定である論理和データベースとする．M は P の最小モデルである．⇔ M は P のモデルであり，かつ各アトム A に関して，$M \models A$ ならば，$P \cup N_P(M) \models A$ が成り立つ．ただし，

$$N_P(M)=\{\neg A \mid \text{アトム } A \text{ は節 } C \in Ground(P) \text{ の頭部に現れ}, M \not\models A\}. \quad (4)$$

□

以下に $MIN(I,Q)$ の計算手順を示す．

(i) I を初期集合として，モデル生成法により $MG^*_P(I) = \{W_1,\ldots,W_n\}$ $(n \geq 0)$ を生成する．ここで，W_i はアトムの集合である．

(ii) 各 $W_i \in MG^*_P(I)$ のうち以下の条件を満たすものをすべて取り出し，その集合を $MIN(I,Q)$ とする．

$$\forall A \in W_i \quad P \cup N_P(W_i) \models A \quad (5)$$

A.2 手続き TPR の正当性

本節では，提案する手続き TPR の正当性（定理 3.1）を示す．

補題 A.2.1 P_{nH} を非ホーン節集合，P_H をホーン節集合，また，P_{neg} を P_H に存在する負節の集合とする．$P_{nH} \cup P_H \models Q$ であり，$P_H \not\models Q$ と仮定する．このとき，$P_H \cup I_i \not\models Q$ なる任意のアトムの集合 I_i について $TRV(P_{neg} \cup \{\leftarrow Q\}, I_i) \neq \phi$ が成り立つ．

証明： $P_{nH} \cup P_H \models Q$ という仮定より，$P_{nH} \cup P_H \cup \{\leftarrow Q\}$ は充足不能である．このとき，この補題は文献 9) 中の Lemma 13 により証明される． □

補題 A.2.1 は，与えられた問合せ Q が $P_{nH} \cup P_H$ のすべての最小モデルにおいて真であり，かつあるモデル候補 I_i に関して $I_i \cup P_H \cup \{\leftarrow Q\}$ のモデルが存在するときには，$(P_{neg} \cup \{\leftarrow Q\}, I_i)$ において全関連であり，かつ違反節であるような非ホーン節が必ず存在することを示している．

次に，補題 A.2.1 に対応する部分関連性についての補題 A.2.3 を証明する．その準備として以下の補題 A.2.2 を証明する．

補題 A.2.2 P_{nH}，P_H，P_{neg} は補題 A.2.1 と同様とする．M を，$M \models Q$ を満たす $P_H \cup P_{nH}$ の最小モデルであるとする．このとき，手続き TPR で生成されるすべてのモデル候補集合 S_i に関して，$M \supseteq I$ なるアトム集合 $I \in S_i$ が存在する．

証明： 帰納法により証明する．

(i) $i=0$ のとき $I=\phi$ であるから明らか．

(ii) 補題 A.2.2 が $i \leq k$ $(k \geq 0)$ で成り立つと仮定する．つまり，$M \supseteq I$ なる I が，S_k 中に存在するとし，以下の 2 つの場合について考える．

　(a) I とは別のモデル候補を拡張することによって S_{k+1} が生成されたとき，I が S_{k+1} にも含まれるので，補題は成り立つ．

　(b) $P_H \cup I$ において違反節である非ホーン節 $C = A_1 \vee \cdots \vee A_m \leftarrow B_1 \wedge \cdots \wedge B_n$ によって I が拡張されたとする．M において C の本体が真となっていることより，C の頭部に M において真となるアトム A'_i が存在する．よって，A'_i を加えることによって生成されるモデル候補 $I \cup \{A'_i\}(\subseteq M)$ が，S_{k+1} に含まれるので，補題は成り立つ． □

補題 A.2.3 P_{nH}，P_H，P_{neg} は補題 A.2.1 と同様とする．問合せ Q が不定であると仮定する．さらに，アルゴリズムの PR 処理部で生成されるモデル候補集

合 $S_i=\{I_1,\ldots,I_m\}$ において以下の式が成り立つとする．

$$\forall I_j \in S_i \quad P_H \cup I_j \not\models Q \tag{6}$$

このとき，あるモデル候補 $I_0 \in S_i$ において $PRV(\{\leftarrow Q\},I_0)\neq\phi$ が成り立つ．

証明： 背理法により証明する．すべてのモデル候補 $I\in S_i$ において，$PRV(\{\leftarrow Q\},I)=\phi$ が成り立つとする．また，仮定より，問合せ Q は不定であるので，$M\models Q$ を満たす $P_{nH}\cup P_H$ の最小モデル M が存在する．さらに，補題 A.2.2 より，$I_0\subseteq M$ を満たす $I_0\in S_i$ が存在することが分かる．

ここで，最小モデル M に関して以下のような確定ホーン節集合を考える．

$$\begin{aligned}&P_{I_0}(M)\\&\;=P_H\\&\;\cup\{A\leftarrow\Gamma \mid A\text{ は節}\Sigma\leftarrow\Gamma\in Ground(P_H\cup P_{nH})\\&\quad\text{の頭部に現れ},\, P_H\cup I_0\not\models\Sigma,\, M\models A\}\\&\;\cup\{A\leftarrow\Gamma \mid A\text{ は節}\Sigma\leftarrow\Gamma\in Ground(P_H\cup P_{nH})\\&\quad\text{の頭部に現れ},\, M\not\models\Sigma\}\end{aligned}$$

すると，M は $P_{I_0}(M)$ のモデルであることが分かる．$P_{I_0}(M)$ の最小モデルを $\mathcal{M}$ とすると，$\mathcal{M}\subseteq M$ が成り立つ．一方，$\mathcal{M}$ は $P_H\cup P_{nH}$ のすべての節を満たすので $P_H\cup P_{nH}$ のモデルであり，M は $P_H\cup P_{nH}$ の最小モデルであるので，結局 $M=\mathcal{M}$ であることが分かる．

もし，$P_{I_0}(M)$ に含まれるある節の頭部に $(\{\leftarrow Q\},I_0)$ に関して部分関連アトムの代入例が存在するならば，部分関連アトムの定義，および $PRV(\{\leftarrow Q\},I_0)=\phi$ より，その節の本体に，少なくとも1つ $(\{\leftarrow Q\},I_0)$ に関して部分関連のアトムが存在することが分かる．つまり，頭部に $(\{\leftarrow Q\},I_0)$ に関して部分関連であるアトムの代入例を持ち，$P_H\cup I_0$ において本体が充足されている節は $P_{I_0}(M)$ には存在しないことが分かる．したがって，$P_{I_0}(M)$ の最小モデルには，$(\{\leftarrow Q\},I_0)$ に関して部分関連アトムの代入例は含まれない．しかし，これは $\mathcal{M}=M\ni Q$ で，Q が $(\{\leftarrow Q\},I_0)$ に関して部分関連アトムであることに矛盾する．□

補題 A.2.4 手続き TPR の TR 処理部で生成されるモデル候補 $I_j\in S_i$ が，以下の条件を満たすとき，問合せ Q は真である．

$$\forall I_j\in S_i\ P_H\cup I_j\cup\{\leftarrow Q\}\models \mathit{false} \tag{7}$$

証明： SATCHMORE[9]の健全性により証明される．□

補題 A.2.5 手続き TPR の PR 処理部で生成されるモデル候補 $I_j\in S_i$ が，以下の条件を満たすとき，問合せ Q は不定である．

$$\exists I_j\in S_i\ MIN(I_j,Q)\neq\phi. \tag{8}$$

証明： 命題 A.1.1[15]および，$MIN(I,Q)$ の定義により，証明される．□

定理 3.1（アルゴリズム $\boldsymbol{TPR}$ の正当性） P を定数記号以外の関数記号を含まず，領域限定である論理和データベースとする．ここで，基底の問合せ Q が与えられたとき，

Q が真	iff	$TPR(P,Q)=$ true
Q が不定	iff	$TPR(P,Q)=$ unknown
Q が偽	iff	$TPR(P,Q)=$ false

が成り立つ．□

証明：

(⇐)

(i) $TPR(P,Q)$ = true とする．このとき，補題 A.2.4 と手続き TPR の定義より，Q は真である．

(ii) $TPR(P,Q)$ = unknown とする．このとき，補題 A.2.1，補題 A.2.5 および手続き TPR の定義より，Q は不定である．

(iii) $TPR(P,Q)$ = false とする．これは，TR 処理部および PR 処理部から抜けたことを意味する．このとき，補題 A.2.1 および，補題 A.2.3 より，Q は真でも不定でもない．よって，Q は偽である．

(⇒) Q が真であるとする．ここで，もし $TPR(P,Q)\neq$ true であったとすると，上記の証明 (⇐) と矛盾する．よって，$TPR(P,Q)$ = true となる．他の場合にも同様に証明できる．□

（平成8年 6月24日受付）
（平成8年10月 1日採録）

島尻 優香（正会員）
1994年名古屋工業大学工学部電気情報工学科卒業．1996年同大学院工学研究科博士前期課程電気情報工学専攻修了．現在，同大学知能情報システム学科助手．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．1985～1989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，人工知能学会理事．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等の研究と開発に従事．電子情報通信学会，人工知能学会，ファジィ学会各会員．