

演繹データベースにおける問合せ処理

ICOT

第3研究室

世木博久

演繹データベース(deductive databases)

知識(事実, ルール) ← 論理式

演繹規則 ← 一階論理

(目的)

- 表現力の拡張

確定ホーン節 → より一般的な論理式へ

- 効率的な問合せ評価アルゴリズム

拡張(否定の導入)

表現力：確定ホーン節データベース → 対象とするクラスの決定

否定情報の扱い [Reiter] → セマンテックスの決定

問合せ評価：トップダウン型アルゴリズム
ボトムアップ型アルゴリズム
融合型 → アルゴリズムの決定

データベースにおける否定情報の扱い

閉世界仮説(Closed World Assumption, CWA) [Reiter]

『データベースから命題 P が証明されないならば、 $\sim P$ と仮定する』

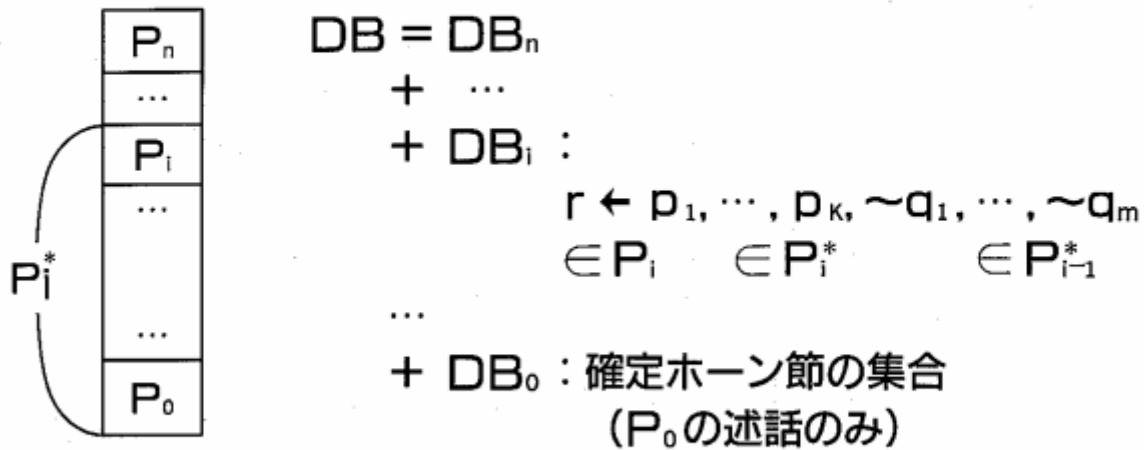
○ 確定ホーン節データベースに対しては適切

× 一般的な論理式を含む場合は矛盾する可能性がある

☆ CWA的な意味論でどこまで拡張できるか？

層状(stratified)データベース [Apt-Blair-Walker]

述語記号 \longrightarrow ルール集合の層化
の分割



- ☆ “ルール的前提部において、下の階層の述語の否定が使える”
- ☆ “ $P \leftarrow \dots, \sim P, \dots$ ” のようなルールは排除される

層状データベースの例：(グラフの接続問題)

$$DB = DB_1 \cup DB_0$$

$$DB_1 = \{ \text{ok}(X, Y) \leftarrow \text{good}(X, Y), \sim \text{poor}(X, Y) \}$$

$$DB_0 = \{ \text{poor}(X, Y) \leftarrow \text{bad}(X, Y).$$

$$\text{poor}(X, Y) \leftarrow \text{bad}(X, U), \text{poor}(U, Y). \}$$

\cup ファクトの集合

\dots ホーン節集合

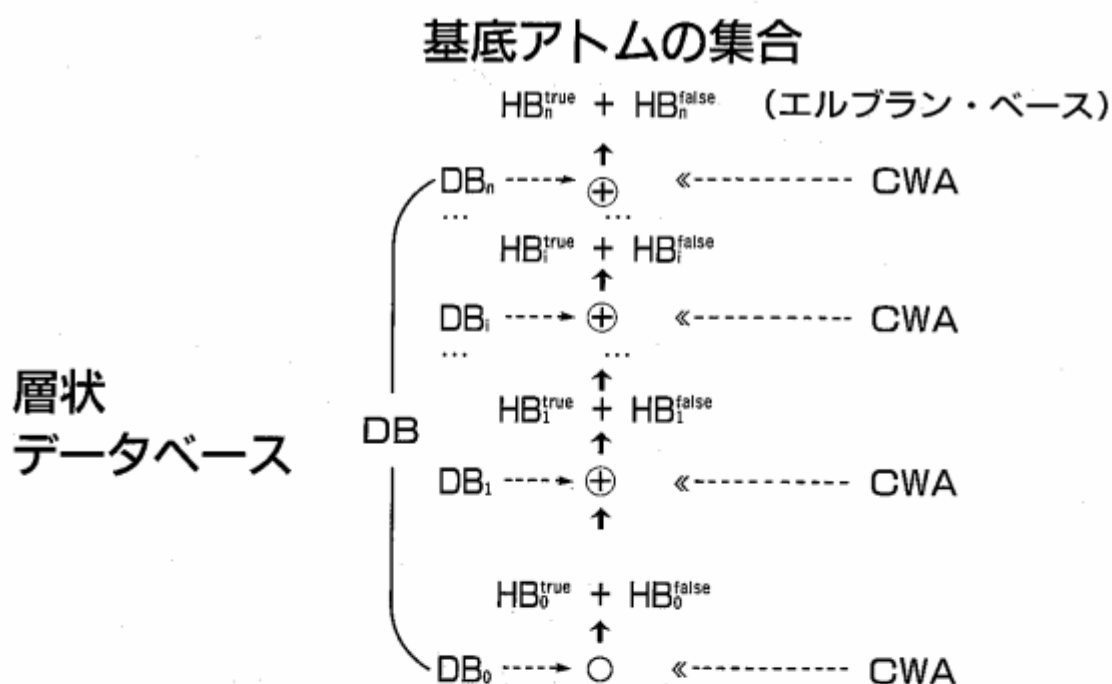
ファクトの集合

$$= \{ \text{各ノード間の bad, good に関するデータ} \}$$



- ☆ “通常ほとんどのデータベースは層状である”

層状データベースのための拡張CWA



“述語の階層に従って, 下から順に閉世界仮説を適用する”

← 確定ホーン節の場合の自然な拡張

問合せ評価アルゴリズム

OLDTNF 反駁

:= OLD反駁 [Tamaki-Sato]

+ Negation as Failure ルール

= OLD反駁 … ホーン節に対する

Prolog の計算モデル

+ Tabulation (テーブル化)

… ゴールの無限展開の抑止

+ Negation as Failure ルール

… 負ゴールの扱い

☆通常のPrologインタプリタの“最小の”拡張

OLDTNF 反駁の例(続き)

s: ←ok(2, Y)
|
←good(2, Y), ~poor(2, Y)
| Y=3
←~poor(2, 3)

“Negation as Failure ルール”

s: ←poor(2, 3)
/ \
←bad(2, 3) ←bad(2, U), poor(U, 3)
x | U=1
s: ←poor(1, 3)
/ \
←bad(1, 3) ←bad(1, U₁), poor(U₁, 3)
x | U₁=2
1: ←poor(2, 3)
有限失敗

問合せ評価アルゴリズム

(対象とするクラスと正当性)

- 対象とする層状データベースのクラス
 - ① 負ゴールを評価するときに、基底項になっている
←通常のPrologの実行の制限と同じ
 - ② ゴールを評価する途中で、出現する項の複雑さが無制限に増大しない
←通常のデータベースで扱われる範囲(Datalog)を含んでいる
- OLDTNF 反駁の正当性
上記の条件の下で、問合せ評価アルゴリズムの健全性、完全性が証明された。

成果と今後の課題

☆成果

| | |
|------------------|------------------------|
| | 否定の導入 |
| 表現力：確定ホーン節データベース | → 層状データベース (のあるクラス) |
| 否定情報の扱い | → 拡張CWA |
| 問合せ評価：トップダウン型 | → OLDTNF 反駁 |
| アルゴリズム ボトムアップ型 | |
| 融合型 | |

☆課題

- アルゴリズムが扱える範囲の拡張
- ボトムアップ型の問合せ処理アルゴリズムの実現