ICOT Technical Memorandum: TM-0400

TM-0400

演繹データベースにおけるコンパイル手法の問い合わせ集合処理への応用

坂間千秋，伊藤英則

October, 1987

ICOT
Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan
(03) 456-3191～5
Telex ICOT J32964

Institute for New Generation Computer Technology

演繹データベースにおけるコンパイル手法の問い合わせ集合処理への応用

坂間千秋　　伊藤英則

（財）新世代コンピュータ技術開発機構

概要

ホーン節集合からなるような演繹データベースにおいて，問い合わせの集合の効率化処理を行うための２つの手法について述べる．最小汎化法は相似な問い合わせが多い場合に有効であり，一方最大汎化法は問い合わせ間に階層関係がある場合に効果的である．

いずれの方法においてもルールのコンパイル手法として遅延評価手法を用いている．

１．はじめに

演繹データベースは通常，大量のファクトの集合からなる外延データベース（extensional database，以下ｅｄｂと略す）と，比較的少量のルールの集合からなる内包データベース（intensional database，以下ｉｄｂと略す）によって構成される．ここで，ファクトとは関係データベースに陽に貯えられているデータであり，ルールとは関数を含まない（function free）ホーン節（特に手続き節）を指す．

このような演繹データベースシステムにおいては，問い合わせは通常ｉｄｂにおいては推論システムによって，またｅｄｂにおいては関係データベースシステムによって評価される．一般に，このような問い合わせの評価方法としては２つの方法が知られている．一方はインタプリタ手法で，他方はコンパイル手法である（[Gall 84]）．

インタプリタ手法では，問い合わせの評価は推論システムと関係データベースシステムによって同時に行われる．即ち，問い合わせの処理が完了するまでｉｄｂとｅｄｂにおいて繰返し評価される．

これに対してコンパイル手法では，問い合わせの評価はｅｄｂに対して遅延実行される．即ち，問い合わせはｉｄｂにおいて予めｅｄｂに対する問い合わせ形式にコンパイルされた後にｅｄｂにおいて評価される．

これらの方法のうち，一般にデータベースが大量である場合はコンパイル手法の方が有効であることが知られている．これは，ｉｄｂを一度にコンパイルしておくことによってｅｄｂに対するアクセスコストを減らすことが出来るからである（[Chak 82]）．実際，[Yoko 84]や[Boc 86]などのシステムはこのような遅延評価手法に基づいて実現されている．

ところが，いま複数の問い合わせを処理する場合を考えると，それぞれの問い合わせを独立にコンパイル処理していたのでは効率が悪い．このような問い合わせ集合の処理を最適化するためには，予めそれぞれの問い合わせの処理過程から共通部分を抽出しておき，その共通な処理は繰返し行わないようにすることが考えられる．例えば，[Chak 86]ではコネクショングラフを用いてこのような最適化を実現している．

本稿では，演繹データベースにおけるコンパイル手法の自然な拡張として，問い合わせ集合の処理の最適化手法について述べる．以下，第２節では演繹データベースにおけるコンパイル手法について述べ，第３節では問い合わせ集合の処理への応用について述べる．最後に，第４節では性能評価ならびに考察を与える．

２．演繹データベースにおけるコンパイル手法

演繹データベースにおける問い合わせのコンパイルとは，与えられた問い合わせをｉｄｂにおいて変換し，関係データベースシステムによってｅｄｂにおいて処理が可能であるような問い合わせに等価変換することをいう．

いま，問い合わせＱが与えられたとき，Ｑはｉｄｂにおける演繹処理によってｅｄｂに対する問い合わせ Ｑidb に変換される．

$$Qidb = e\ (idb,\ Q) \qquad (1)$$

ここで，c（idb，Q）は idb における問い合わせの演繹処理を表す．

このような変換の後，Qidb は edb において関係演算処理される．

$$Qidb,cdb = r\ (edb,\ Qidb) \qquad (2)$$

ここで，r（edb，Qidb）は Qidb の edb における関係演算処理を表し，Qidb,cdb は問い合わせに対する解集合である．

このような問い合わせの処理過程における問題点は，それが特に再帰処理を含むような場合の停止性と完全性である．実際，このような再帰問い合わせのコンパイル手法に関する多くの研究が成されているが（[Ban 86]），本稿ではこのような再帰問い合わせの処理手法としてはホーン節変換法（[Miya 86]）を用いる．ホーン節変換法とは与えられた問い合わせを edb において定義されている述語と cdb において繰返し処理される再帰述語のみから定義される問い合わせに等価変換する手法である．

［例２．１］

以下のような idb を考える．

```
p(X,Y):- q(X,Z),r(Z,Y).
q(X,Y):- s(X,Y).
q(X,Y):- s(X,Z),t(Z,Y).
t(X,Y):- u(X,Z),q(Y,Z).
v(X,Y):- w(Y,X).
```

ホーン節変換法を使うと，問い合わせ ?- p(a,Y) は idb において以下のように変換される．

```
Qidb
  p(a,Y):- q(a,Z),r(Z,Y).
  q(X,Y):- s(X,Y).
  q(X,Y):- s(X,Z),u(Z,W),q(Y,W).
```

ここで，r，s，u は cdb で定義されている述語で，q は再帰述語である．これらの Qidb は edb において処理される． □

上の例で，Qidb における２番目と３番目の節は q に対する再帰処理のために変数に対して値がバインドされていないことに注意されたい．

３．問い合わせ集合処理への応用

いま，ある演繹データベースに対して処理すべき問い合わせ集合が与えられたとき，前節で述べた手法を繰返し用いて各々の問い合わせを独立に処理するのは効率が悪い．以下ではこのような問い合わせ集合の最適化処理方法について述べる．

３．１　汎化問い合わせ

先ず，以下の議論で用いる幾つかの用語の定義を与える．

［定義３．１］

1．変数，定数は項である．また項はこれらに限る．（先に述べたように，項は関数記号を含まないとする．）

2．p が n 引数述語で t1,..,tn が項であるとき，p（t1,..,tn）はアトム（atomic formula）である．

特に，同じ述語で引数の数が等しいアトムは相似（compatible）であるという． □

［定義３．２］

1．アトム P がアトム Q より汎化されている（more general）とは，ある代入 θ が存在して Pθ＝Q が成り立つことである．このとき，$Q \sqsubseteq P$ と書く．ここで，$\sqsubseteq$ はアトム間の半順序関係である．

2．S をアトムの集合とするとき，

i) $\forall T \in S,\ T \sqsubseteq L$

ii) $\forall Li\ (\ (\forall T \in S \wedge T \sqsubseteq Li) \supset L \sqsubseteq Li)$

を満たすアトム L を S の最小汎化（least generalization）と云い，

iii) $\forall T \in S,\ T \sqsubseteq L$

iv) $\forall Li\ (\ (\forall T \in S \wedge T \sqsubseteq Li) \supset Li \sqsubseteq L)$

を満たすアトム L を S の最大汎化（most generalization）と云う． □

［例３．１］

アトムの集合 { p(a,a)，p(a,b) } の最小汎化，最大汎化はそれぞれ p(a,X)，p(X,Y) である． □

このような汎化手法は，これまでに例えば帰納推論（[Plot 70]）やＯＲ並列探索（[Fish 75]）などに応用されてきたが，以下では演繹データベースにおける問い合わせ集合の処理への適用について述べる．

３．２ 最小汎化法

以下の議論においては，問い合わせは単一のアトムからなるものとする．例えば，複数のアトムからなる問い合わせ ?- p(X,Y),q(Y,Z),r(Z) は単一のアトムからなる問い合わせ ?- s(X,Y,Z) と節 s(X,Y,Z):- p(X,Y),q(Y,Z),r(Z) に分解して考える．

先ず，最小汎化問い合わせを定義する．

［定義３．３］

問い合わせの集合が与えられたとき，その中の相似な問い合わせの最小汎化をそれらの最小汎化問い合わせ（least generalized query）と云い，全ての最小汎化問い合わせの集合をもとの問い合わせ集合に対する最小汎化問い合わせ集合（least generalized query set）と云い，ＬＧＱと書く． □

［例３．２］

問い合わせ集合 {p(a,a), p(b,b), q(a,Y), q(X,b), r(a) } の ＬＧＱ は { p(X,X), q(X,Y), r(a) } である． □

いまある問い合わせ集合が与えられたとき，ＬＧＱは以下の手順によって求められる．

1. 問い合わせ集合を，相似な問い合わせからなる部分集合に分類する．
2. それぞれの部分集合に対して，最小汎化問い合わせを求める．

ここで，最小汎化を求めるアルゴリズムは[Plot 70]，或いは[Rey 70] に依る．

さて，以下ではこのような ＬＧＱ を用いた問い合わせ処理の最適化手法について述べる．

いま，問い合わせ集合 Ｑ が与えられたとする．このとき，ＬＧＱ は Ｑ の上限として以下のように表される．

$$LGQ = \bigsqcup Q \qquad (3)$$

次に，このような ＬＧＱ の各々の要素を ｉｄｂ において変換することを考える．即ち，

$$LGQidb = c\ (idb,\ LGQ) \qquad (4)$$

この結果，ＬＧＱidb は Ｑ の選択条件のドで ｅｄｂ において以下のように評価される．

$$Qidb,rdb = r\ (edb,\ \sigma Q\ (LGQidb)) \qquad (5)$$

ここで，σＱ は Ｑ の条件による選択操作を表し，Ｑidb,rdb はもとの問い合わせ集合に対する解集合を表す．

［例３．３］

例２．１と同じ ｉｄｂ において以下のような問い合わせ集合 Ｑ を考える．

Ｑ = { p(b,Y), p(X,d), t(X,c), t(b,c), v(X,f) }

このとき，Ｑ に対する ＬＧＱ は

ＬＧＱ = { p(X,Y), t(X,c),v(X,f) }

で，それぞれの要素は ｉｄｂ において以下のように変換される．

```
LGQidb
  p(X,Y):- q(X,Z),r(Z,Y).
  q(X,Y):- s(X,Y).
  q(X,Y):- s(X,Z),u(Z,W),q(Y,W).
  t(X,Y):- u(X,Z),s(Y,Z).
  t(X,Y):- u(X,Z),s(Y,W),t(W,Z).
  v(X,f):- w(f,X).
```

ここで，ＬＧＱidb は ｉｄｂ における ＬＧＱ のそれぞれの要素の変換結果の集合和で，与えられたそれぞれの問い合わせにおける選択条件 σX=bvY=d (p(X,Y)), σ(X=b,Y=c)vY=c (t(X,Y)), σY=f (v(X,Y)) を用いて ｅｄｂ において処理される． □

３．３ 最大汎化法

ここでは，問い合わせ集合の最適化処理法として最大汎化法について述べる．

先ず，最大汎化問い合わせを定義する．

［定義３．４］

問い合わせの集合が与えられたとき，その中の相似な問い合わせの最大汎化をそれらの最大汎化問い合わせ（most generalized query）と云い，全ての最大汎化問い合わせの集合をもとの問い合わせ集合に対する最大汎化問い合わせ集合（most generalized query set）と云い，ＭＧＱと書く． □

［例３．４］
問い合わせ集合 {p(a,a), p(b,b), q(a,Y), q(X,b), r(a) } の ＭＧＱ は { p(X,Y), q(X,Y), r(X) } である． □

与えられた問い合わせ集合から ＭＧＱ を求める手順は ＬＧＱ の場合で最小汎化アルゴリズムを使う代わりに，単に問い合わせ中の引数に異なる変数を割り当てることによって得られる．

次に，述語間の半順序関係を定義する．

［定義３．５］
与えられたホーン節集合における述語間の半順序関係を以下のように定義する．
1．ホーン節集合 Ｓ，及び節 Ck ∈ Ｓ とする．
いま，Ck 中の述語 pi，pj（pi ≠ pj）が
pi ∈ Ｈｅａｄ（Ck），pj ∈ Ｂｏｄｙ（Ck）
を満たすとき pi は pj より高位（higher）である（あるいは pj は pi より低位（lower）である）と云い，pj ≦ pi と書く（但し，Ｈｅａｄ（Ck），Ｂｏｄｙ（Ck）はそれぞれ節 Ck のヘッド部，ボディ部に表われる述語の集合を表す）．
2．pi ≦ pj かつ pj ≦ pi のとき pi と pj は同位(similar) であると云い，pi ～ pj と書く．
□

［例３．５］
ホーン節集合
Ｓ＝ { p(X,Y):- q(X,Z),p(Z,Y).
q(X,Y):- r(X,Z),s(Z,Y).
s(X,Y):- q(Y,X). }
において r ≦ q ≦ p ，q ～ s である． □

いま，ｉｄｂ がホーン節集合で与えられたとき，上の定義によって ｉｄｂ 中に表われる述語間に半順序関係が定義される．そこで，このような ｉｄｂ に対して問い合わせ集合が与えられたとき，それぞれの問い合わせの述語間に ｉｄｂ に基づいた順序関係が定義される場合がある．なぜなら，問い合わせにおける述語は通常 ｉｄｂ 中に定義されているからである．

さて，以下では上で導入した ＭＧＱ，及び述語間の半順序関係を用いた問い合わせ処理の最適化手法について述べる．
いま，問い合わせ集合 Ｑ が与えられたとする．このとき，Ｑから定まる ＭＧＱ において述語間に順序関係が定義されているとき，ＭＧＱ の要素をこの順序関係に基づいてソートすることを考える．

$$s(MGQ) = < mgq1, \ldots, mgqn > \quad (6)$$

ここで，ｓ（ＭＧＱ）はソートされた ＭＧＱ で，$1 \leqq i < j \leqq n$ のとき ｍｇｑｉ は ｍｇｑｊ よりも低位の最大汎化問い合わせである．
次に，このようにソートされた ＭＧＱ は，ｉｄｂ において低位のものから順に評価する．なぜなら，高位の述語は ｉｄｂ 中でより低位の述語によって定義されているため，低位の問い合わせの評価結果はより高位の問い合わせの評価過程において使うことが出来るからである．
ここで注意しておかねばならないことは，高位の問い合わせは低位の問い合わせ述語に対して，より一般的な処理結果を必要とすることがあるので，予め低位の問い合わせを最大汎化して処理しておく必要があるのである．

さて，こうしてソートされた ＭＧＱ の ｉｄｂ における評価過程は以下のように表される．

$$s(MGQ)idb = c(idb, s(MGQ)) \quad (7)$$

ここで，

$$c(idb, s(MGQ)) = \cup i\ c(idbi, mgqi),$$

$$idbi = c(idbi\text{-}1, mgqi\text{-}1) \cup idb'i\text{-}1 \quad (i \geqq 2),$$

$$idb1 = idb$$

で，ｉｄｂ'i-1 は ｉｄｂi-1 から ｍｇｑi-1 の評価結果と同じ述語をヘッド部に持つ節を除いたものである．即ち，ｍｇｑｉ は ｉｄｂ においてそれよりも低位の ｍｇｑ の評価結果を用いて評価される．なお，∪ｉ ｃ（ｉｄｂｉ，ｍｇｑｉ）はソートされたＭＧＱ の ｉｄｂ における評価結果の集合和である．
この結果，ｓ（ＭＧＱ）idb は Ｑ の選択条件の下で ｅｄｂ において以下のように評価される．

$$Qidb,edb = r(edb, \sigma Q(s(MGQ)idb)) \quad (8)$$

ここで，σＱ は Ｑ の条件による選択操作を表し，Ｑidb,edb はもとの問い合わせ集合に対する解集合を表す．

[例３．６]

次のような ｉｄｂ と問い合わせの集合 Q を考える．

```
p(X,Y):- q(X,Z),r(Z,Y).
q(X,Y):- s(X,Y).
q(X,Y):- s(X,Z),t(Z,Y).
r(X,Y):- t(X,Y),w(Y,Z).
t(X,Y):- u(X,Z),q(Y,Z).
```

Q = {p(a,Y), p(a,b), q(X,c), q(b,Y), r(X,f) }

このとき，Q に対する ＭＧＱ は

ＭＧＱ＝ { p(X,Y), q(X,Y), r(X,Y) }

で，ｉｄｂ 中の述語に関する順序関係にしたがって以下のようにソートされる．

s(ＭＧＱ) =< (q(X,Y) r(X,Y)) , p(X,Y) >

ここで q ≦ p，r ≦ p で，q と r の間には順序関係はない．

そこで，先ず q(X,Y) と r(X,Y) が ｉｄｂ において以下のように評価される．

```
q(X,Y):- s(X,Y).
q(X,Y):- s(X,Z),u(Z,W),q(Y,W).
r(X,Y):- u(X,W),q(Y,W),w(Y,Z).
```

この結果，ｉｄｂ は以下のように変換される．

```
idb1
p(X,Y):- q(X,Z),r(Z,Y).
q(X,Y):- s(X,Y).
q(X,Y):- s(X,Z),u(Z,W),q(Y,W).
r(X,Y):- u(X,W),q(Y,W),w(Y,Z).
t(X,Y):- u(X,Z),q(Y,Z).
```

次に，このような ｉｄｂ1 において p(X,Y) が以下のように変換される．

```
p(X,Y):- q(X,Z),u(Z,W),q(Y,W),w(Y,U).
```

この結果，ｉｄｂ におけるＭＧＱ の評価結果はそれぞれの結果の集合和として以下のようになる．

```
s(ＭＧＱ)idb
p(X,Y):- q(X,Z),u(Z,W),q(Y,W),w(Y,U).
q(X,Y):- s(X,Y).
q(X,Y):- s(X,Z),u(Z,W),q(Y,W).
r(X,Y):- u(X,W),q(Y,W),w(Y,Z).
```

これらは最後に Q における選択条件 σ(X=a,Y=b)∨X=a (p(X,Y))，σX=b∨Y=c (q(X,Y))，σY=f (r(X,Y)) を用いて ｅｄｂ において処理される．　□

上の例では，Q における p(a,Y)，p(a,b) については実は最大汎化を行う必要はない．なぜなら，この場合これらの問い合わせの処理結果を使うような上位の問い合わせが，与えられた問い合わせ集合の中に含まれていないからである．一般に与えられた問い合わせ集合のうち，最上位のものについては最小汎化を行えば十分である（この場合は p(a,Y) となる）．

４．評価，考察

前節で，演繹データベースにおける問い合わせ集合の最適化処理方法として，汎化を用いた２つの方法について述べた．図１にこれらの方法の比較を示す．

いま，与えられた問い合わせ集合において相似な問い合わせが多い場合は，ＬＧＱ法が有効である（ａ）．この場合，それぞれの問い合わせの探索空間はＬＧＱのそれによって近似することが出来る．

一方，問い合わせ集合において階層的順序関係を持つ問い合わせが多い場合は，ＭＧＱ法が有効である（ｂ），この場合，それぞれの問い合わせの探索空間は下位のものから順に縮めていくことが出来る．

しかし，（ｃ）のような場合は，それぞれの問い合わせが共通の探索空間を持っているにも拘らず，どちらの方法もあまり効果的ではない．これは，それぞれの問い合わせを展開してみないとその共通部分が検出出来ないからである．このような場合は，例えばボトムアップ的解析する方法などが考えられるが，本稿では触れないことにする．

さて，以下では ＬＧＱ法と ＭＧＱ法を使った問い合わせ処理の性能評価について述べる．

ここではＬＧＱ，ＭＧＱ 処理系，及びホーン節変換処理系は DEC-10 Prolog 上でインプリメントされたものを使った．表１は ＬＧＱ処理，及びＭＧＱ処理の実行時間を示している．ここで，ＬＧＱ処理とは問い合わせ集合から ＬＧＱ を求める処理を指し，Ｍ

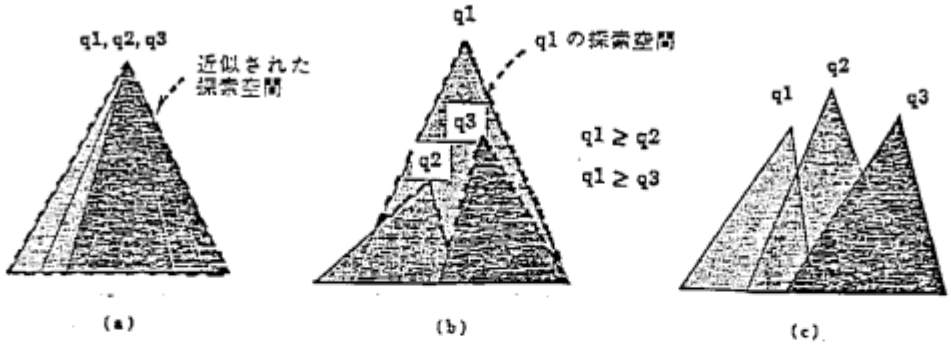

図１　　q1, q2, q3：問い合わせ

ＧＱ処理とは問い合わせ集合から ＭＧＱ を求め，それを述語の順序関係に従ってソートする処理を指す．

(a) LGQ/Q = MGQ/Q = 0.6

問合せ数	5	10	15
LGQ	15	42	82
MGQ	14	36	74

(msec)

(b) 問合せ数：10

/Q	0.2	0.4	0.6	0.8
LGQ	31	37	43	44
MGQ	16	28	39	45

(msec)

表１

この実験では，問い合わせは関数を含まない２項関係を仮定している．表１(a) は問い合わせの数が最小汎化，最大汎化によって６０％減の場合の，異なる問い合わせの数に対するそれぞれのコンパイル実行時間を示している．また，表１(b) は問い合わせの数が１０個の場合の，異なる汎化の割合に対するコンパイル実行時間を示している．

次に，これらの方法を用いた ｉｄｂ における問い合わせ処理の性能効率を示す．実験では，関数と定数を含まない２項関係によって構成されたホーン節集合からなり，全体の６０％が線型再帰節 (linear recursive) であるような ｉｄｂ を仮定した．一つの問い合わせに対するホーン節変換のコンパイル実行時間を表２に示す．ここで仮定した ｉｄｂ では，深さが増すにつれて探索空間が指数関数的に拡がるため，実行時間もそれに従って増えている．

深さ	5	10	15
処理時間	863	5224	12647

(msec)

表２

さて，このような ｉｄｂ において問い合わせ集合を処理するに当たって，以下の３つの方法で評価した．

① 各々の問い合わせを独立に，繰り返し評価する．
② ＬＧＱ 法を用いる．
③ ＭＧＱ 法を用いる．

汎化比６０％ (LGQ/Q = MGQ/Q = 0.6) のときの，問い合わせ数が５個，及び１０個の場合のコンパイル実行時間の比較を図２に示す．

この実験で使ったサンプル ｉｄｂ は階層的なものであったため，深さが増すにつれて ＭＧＱ 法が他の２つの方法に較べて効率が上がっている．

なお，この実験では定数を含まない ｉｄｂ を仮定したため，問い合わせの汎化による探索空間の拡がりが全体の評価に影響を及ぼさなかったが，実際 ｉｄｂ 中に定数が含まれていて，汎化によって探索空間が拡がってしまうような場合には，ここで述べた方法による処理効率も落ちることは考えられる．

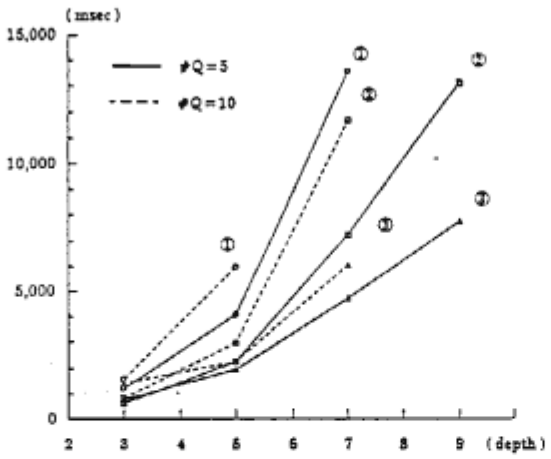

図2

① 繰り返し法

② LGQ 法

③ MGQ 法

(LGQ/Q = MGQ/Q = 0.6)

5. おわりに

本稿では，演繹データベースにおける問い合わせ集合の最適化処理方法について述べた．汎化を用いた２つの方法は，従来の ｉｄｂ における問い合わせのコンパイル法の自然な拡張であり，問い合わせ集合のコンパイルのコスト減に有効であると考えられる．また，本稿では関数を含まないホーン節集合を仮定したが，これらの汎化手法は一般に関数を含む場合にも適用が可能である．

今後は，アプリケーション分野に応じた検討が必要と考えている．

謝辞

本稿を纏めるに当たり，森田幸伯，宮崎収兄各氏を始めとするＩＣＯＴにおけるＫＢＭプロジェクトの諸兄との議論は非常に有効であった．また，評価実験における武脇敏見氏のサポートにも併せて感謝したい．

参考文献

[Ban 86] Bancilhon,F., et al.: "An Amateur's Introduction to Recursive Query Processing Strategies", Proc. ACM SIGMOD 86, pp.16-52, 1986.

[Boc 86] Bocca,J., et al.: "Some Steps Towards a DBMS Based KBMS", Proc. Information Processing Congress, pp.1061-1067, 1986.

[Chak 82] Chakravathy,U.S., et al.: "Interfacing Predicate Logic Languages and Relational Databases", Proc. 1st ICLP, pp.91-98, 1982.

[Chak 86] Chakravathy,U.S., et al: "Multiple Query Processing in Deductive databases using Query Graphs", Proc. 12th VLDB, pp.384-391, 1986.

[Fish 75] Fishman,D.H., et al: "H-Representation: A Clause Representation for Parallel Search", Artificial Intelligence, vol.6, pp.103-127, 1975.

[Gall 84] Gallaire,H., et al.: "Logic and Databases: A Deductive Approach", ACM Computing Surveys, vol.16, No.2, pp.153-185, 1984.

[Miya 86] Miyazaki,N., et al.: "Compiling Horn Clause Queries in Deductive Databases: A Horn Clause Transformation Approach", TR-183, ICOT, 1986.

[Plot 70] Plotkin,G.D.: "A Note on Inductive Generalization", Machine Intelligence, vol.5, pp.153-163, 1970.

[Rey 70] Reynolds,J.C.: "Transformational Systems and the Algebraic Structure of Atomic Formulas", Machine Intelligence, vol.5, pp.135-151, 1970.

[Yoko 84] Yokota,H., et al: "An Enhanced Inference Mechanism for Generating Relational Algebra Queries", Proc. 3rd ACM PODS, pp.229-238, 1984.