データベースシステム 124−21
情 報 学 基 礎 62−21
（２００１．５．２２）

XMLデータベースの問合せ最適化に関する検討

加藤弘之

国立情報学研究所

XMLデータベースに対する論理レベルでの問合せ最適化の必要性について記述する．まず，先日W3CよりWorking Draftとして発表されたXML問合せ代数レベルでの問合せ最適化を試み，代数レベルだけではなく論理レベルでの問合せ書換えによる最適化手法の必要性とその手法について検討する．

A Query Optimization for XML Databases

Hiroyuki KATO

National Institute of Informatics

This paper describes that it is necessary to develop a query optimization in logcal level. After trying to optimize queries in the XML Query Algebra submitted by W3C as a Working Draft, a query optimization in logical level by means of rewriting queries is shown. Also, a logic for querying XML data is considered to apply a query optimization by means of query rewrite.

1. はじめに

XML[WWWC98] は W3C(World Wide Web コンソーシアム) によって勧告された，内部に構造を有するデータを記述するためのメタ言語である．XML の重要な応用の一つが Web 上のデータの共通交換フォーマットである．異種分散環境における情報源のデータを XML でラップすることで，格納手法に依らない情報の統合が可能となる．

このような異種分散環境における情報源のデータの統合はデータベースビューとして捉えることができる．全ての情報源が問合せ最適化機構を支援しているわけではないので，XML データベースに対する問合せ最適化手法を開発することは有益なことである．

本稿では，先日 W3C より Working Draft として発表された XML 問合せ代数 [WWWC01a] による問合せ最適化を試みる．そして，代数レベルではなく論理レベルでの問合せ最適化が必要であることを示す．本稿は以下のような構成である．第 2 節では XML 問合せ代数を用いた問合せとその代数レベルでの最適化について述べる．第 3 節では代数レベルの問合せだけでは不十分であり，論理レベルでの最適化の必要性について述べる．第 4 節は結論と今後の課題である．

2. XML 問合せ代数

XML 問合せ代数は，XQuery[WWWC01b] に対して操作的意味を提供しているものである．この節では XML 問合せ代数による問合せの記述とこの代数レベルでの問合せ最適化について，[FSW01] に基づいて記述する．

2.1 問合せの代数表現

図 1と図 2に XML 問合せ代数モデルにおけるスキーマ例とデータ例を示す．基本的に [WWWC01a, FSW01] で用いられているものと同じものであるが，`book` の子エレメントとして`keywd` を新たに追加したものとなっている．

このようなデータに対して，「logic programming について多くの本を書いた著者による XML に関する本を検索する．」というのは十分に現実的な問合せである．この問合せは次のように翻訳することができる．

```
type Bib =
 bib [Book*]
type Book =
 book [
  title  [String],
  year   [Integer],
  author [String]+,
  keywd  [String]*
]
```

図 1 XML 問合せ代数モデルにおけるスキーマ例

```
let book0 : Bib =
 bib [
  book [
    title  ["Data on the Web"],
    year   [1999],
    author ["Abiteboul"],
    author ["Buneman"],
    author ["Suciu"],
    keywd  ["XML"],
    ...
  ],
  book [
    title  ["Logic Programming
               and  Dababases"],
    year   [1990],
    author ["Ceri"],
    author ["Gottlob"],
    author ["Tanca"],
    keywd  ["logic programming"],
    ...
  ],
...
 ]
```

図 2 XML 問合せ代数モデルにおけるデータ例

```
for a in distinct(bib0/book/author) do
 for k1 in (for b1 in bib0/book do b1/keywd) do
  where k1/data() = "logic programming" do
   for a1 in b1/author do
    where a/data() = a1/data() do
     where count(b1) > ( for aa in distinct(bib0/book/author) do
                          for k2 in (for b2 in bib0/book do b2/keywd) do
                           where k2/data() = "logic programming" do
                            for a2 in b2/author do
                             where aa/data() = a2/data() do
                              avg(count(b1)) )
      for k3 in (for b3 in bib0/book do b3/keywd) do
       where k3/data() = "XML" do
        for a3 in b3/author do
         where a3/data() = a/data() do
          b3
```

図 3 XML 問合せ代数による問合せ例

Q: keywd に"logic programming" が出現する本を平均以上書いた著者によるkeywd に"XML" が出現する本の検索

この問合せの XML 問合せ代数による記述を図3に示す[1]

2.2 XML 問合せ代数における最適化

XML 問合せ代数におけるfor 式による繰り返しの構造はモナドの構造に相当する．入れ子関係代数においてこのモナド法則が最適化に用いられたように，XML 問合せ代数においても，モナド法則による最適化が提案されている．

三つのモナド法則は次の通りである．

- left unit law
  for v in e_1 do $e_2 = e_2 v := e_1$
- right unit law
  for v in e do $v = e$
- associative law
  for v_2 in (for v_1 in e_1 do e_2) do e_3
  $=$ for v_1 in e_1 do (for v_2 in e_2 do e_3)

問合せ例 **Q** においては，上記のモナド法則のうち assiciative law が適用可能である．図 4 に associative law を適用した結果を示す．

3. 論理レベルでの問合せ最適化の必要性

代数レベルでの問合せ最適化とは，手続き型言語のコンパイル時における最適化処理に類似した点がある．宣言的な言語レベルでの最適化，すなわち，問合せの書き換えによる最適化は，より効率的な最適化を実現することができる．

例えば，図 4に示した問合せは,"logic programming" をkeywd に含む本の数と"logic programming" を含む本の平均を比較するところで，非常に冗長な処理をしている．したがって，この問合せには未だ最適化の余地がある．例えば，まず"logic programming" を keywd に含む本をビューとして計算しておき，このビューを利用して問合せを書き換えた例を図 5に示す．

問題はこのような問合せ書換えを機械的に行なうことができるかである．実は，この書換えはマジックという手法を用いることで，機械的に問合せ書換えができる可能性がある．

[1]XML 問合せ代数には関係代数における Group-By([Ram97]) に相当する高階 (二階) の関数は用意されていない (XQuery には用意されている．)．その代わりにdistinct とfor を組み合せて対応することができる．尚，distinct は ICDT version[FSW01] であり，WD[WWWC01a] ではdistnct_value となっている．

```
for a in distinct(bib0/book/author) do
 for b1 in bib0/book/ do
  for k1 in b1/keywd do
   where k1/data() = "logic programming" do
    for a1 in b1/author do
     where a/data() = a1/data() do
      where count(b1) > ( for aa in distinct(bib0/book/author) do
                            for b2 in bib/book do
                             for k2 in b2/keywd do
                              where k2/data() = "logic programming" do
                               for a2 in b2/author do
                                where aa/data() = a2/data() do
                                 avg(count(b1)) )
       for b3 in bib0/book do
        for k3 in b3/keywd do
         where k3/data() = "XML" do
          for a3 in b3/author do
           where a3/data() = a/data() do
            b3
```

図 4 **Q1** への結合法則の適用結果

3.1 マジック

マジックは，演繹データベースの分野において，論理プログラミングにおける手法を採り入れた再帰問合せのための最適化手法として，当初開発された [BMSU86, BR87]. その手法は，set-at-a-time に実行する SLD-resolution(Selection rule-driven Linear resolution for Definite clauses) をシミュレートしたホーン節をボトムアップに解くものである [AHV95]. ボトムアップアプローチを採用することで，処理の停止が保証されているばかりか，トップダウンアプローチをシミュレートするので，問合せ処理中において，答えに直接結びつかない余分な組の生成を抑えることができる [Ull89]. 様々な亜種が考え出されたが，より複雑な問合せにも対応できる Supplementary Magic が一般的なものである [AHV95]. これは supplementary 関係を用いて SIPS(Sideways Information Passing Strategy) を実現する点にある. その後，非再帰問合せに適用され，関係データベース問合せの最適化手法としても確立されたものとなった [Mum91, SHP+96].

このマジックを XML 問合せに適用するためには，適用する論理を決定する必要がある. したがって，XML 問合せ言語 XQuery に対して表示的意味を与えることのできる，well-defined な XML 問合せ論理の開発が必要である.

4. 結論と今後の課題

本稿では，XML データベースに対する問合せについて代数レベルの最適化だけでは十分ではなく，論理レベルでの最適化も必要であることを示した. そのために，well-defined な XML 問合せ論理の開発が必要である.

XML データの単純な論理表現として，ある内容モデルについて左辺を述語名とし右辺を属性名とするような述語論理で記述することが可能である. ここで，XML 問合せ言語ではエレメント名を返す問合せを記述することができる. この特徴をそのまま適用してしまうと高階論理となってしまう. しかしながら，高階な文法を有しながらその意味は一階に抑えている論理として HiLog[CKW93] がある. HiLog の意

```
let lbook = for b in bib0/book do
              for k in b/keywd do
                where k/data() = "logic programming" do b

let lauth = for a in distinct(lbook/book/author) do   a

let lbooknumbyauth = for a1 in lauth do
                       numbyauth [a1,
                       for b in lbook/book do
                         for a2 in b/author do
                           where a1/data() = a2/data() do num[count(b)]]

let lbookavg = for n in numbyauth do  avg(num/data())

for a in lauth do
  for b in bib/book do
    for k in b/keywd do
      where k/data() = "XML"
        for a1 in b/author do
          where a/data() = a1/data() do
            for lbn in lbooknumbyauth do
              where lbn/author/data() = a1/data() do
                for avgnum in lbookavg do
                  where lbn/num/data() > avgnum/data() do b
```

図 5 問合せ書き換えによる最適化の例

味は一階であるので，健全で完全な証明手続きが存在する．

HiLog の特徴は項が原子式でもある点にある．したがって，HiLog においてエルブラン空間とエルブラン基礎が一致する．これによりエルブラン解釈はエルブラン空間 (エルブラン基礎) の部分集合となる．このような特徴を有する HiLog のストラクチャは，通常の一階述語論理のストラクチャと異なり，四つ組 $<U, U_{true}, I, \mathcal{F}>$である．但し，

- Uは，$\mathbf{M}$ の定義域の内包の集合で非空である．
- U_{true}は，Uの部分集合で真の命題の内包である．
- $I : \mathcal{S} \mapsto U$は内包と各論理記号を結び付ける関数．
- $\mathcal{F} : U \mapsto \prod_{k=1}^{\infty}[U^k \mapsto U]$ は関数．
  但し，for every $u \in U$, $k \geq 1$, $\mathcal{F}(u)$ の k番目に関する射影 (以下，$u_{\mathcal{F}}^{(k)}$で表す．) は，$[U^k \mapsto U]$ 中のある関数である．$\prod$は集合の直積を表し，$[U^k \mapsto U]$ は，全ての k-ary 関数 $(U^k \mapsto U)$ の集合を表す．

また，変数割り当て$\nu : \mathcal{V} \mapsto U$は，以下のように項の集合 $\mathcal{T}$へと拡張されている．ストラクチャ$\mathbf{M}$ が与えられたとき，

- 任意の $s \in \mathcal{S}$に対して，$\nu(s) = I(s)$
- $\nu(t(t_1, ..., t_n)) = (\nu(t))_{\mathcal{F}}^{(n)}(\nu(t_1), ..., \nu(t_n))$

したがって，HiLog 項 $t(t_1, ..., t_k)$ の解釈は以下のようになる．

- k-ary 関数の場合
  変数割り当てνの下での外延的意味は，tの内包に関連づけられた関数のベクトルのk番目の要素である．すなわち，$(\nu(t))^{(k)}_{\mathcal{F}}$
- 原子式の場合
  真となるのは，iff. $\nu(t(t_1,...,t_k)) \in U_{true}$

今後の課題はこのHiLogに対してマジックを適用することである．これにより，XMLデータベースに対する論理レベルでの問合せ最適化手法の基礎を与えることができる．

参考文献

[AHV95] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases.* Addison-Wesley, 1995.

[BMSU86] F. Bancilhon, D. Maier, Y. Sagiv, and J. Ullman. Magic sets and other strange ways to implement logic programs. In *Proc. ACM SIGACT-SIGMOD Symp. on Principles of Database Sys.*, Boston, MA, 1986.

[BR87] C. Beeri and R. Ramakrishnan. On the power of magic. In *Proc. ACM SIGACT-SIGMOD Symp. on Principles of Database Sys.*, p. 269, San Diego, CA, March 1987.

[CKW93] W. Chen, M. Kifer, and D. S. Warren. HiLog: A foundation of higher-order logic programming. *The Journal of Logic Programming*, Vol. 15, No. 3, pp. 187–230, 1993.

[FSW01] M. Fernandez, J. Simeon, and P. Wadler. A semi-monad for semi-structured data(icdt version). In J. Van den Bussche and V. Vianu, editors, *Proc. International Conference on Database Theory(ICDT)*, pp. 263–300, 2001.

[Mum91] I. S. Mumick. *Query optimization in deductive and relational databases.* PhD thesis, Stanford University, 1991.

[Ram97] R. Ramakrishnan. *Database Management Systems.* McGraw-Hill, 1997.

[SHP+96] P. Seshadri, J. M. Hellerstein, H. Pirahesh, T. Y. C. Leung, R. Ramakrishnan, D. Srivastava P. J. Stuckey, and S. Sudarshan. Cost-based optimization for magic: Algebra and implementation. In *Proc. ACM SIGMOD International Conference on Management of Data*, pp. 435–446, June 1996.

[Ull89] J. D. Ullman. *Principles of Database and Knowledge-base Systems*, Vol. II. Computer Science Press, 1989.

[WWWC98] World Wide Web Consortium. eXtensible Markup Language (XML) 1.0. http://www.w3.org/TR/1998/REC-xml-19980210, February 1998. W3C Recommendation 10-February-1998.

[WWWC01a] World Wide Web Consortium. The XML Query Algebra. http://www.w3.org/TR/query-algebra, February 2001. W3C Working Draft 15 February.

[WWWC01b] World Wide Web Consortium. XQuery: A Query Language for XML. http://www.w3.org/TR/xquery, February 2001. W3C Working Draft 15 February.