複雑問い合わせに対する効率的なジェネラル・フィルタリング

3R－1

陳 漢雄†　大保 信夫‡

†つくば国際大学 産業情報学科 ‡筑波大学 電子情報工学系

1　はじめに

近年、データベースシステムの様々な分野への応用にともない、アプリケーションのオブジェクトが巨大化になりつつあり、当然ながら問い合わせの処理も複雑になっている。評価に非常に時間がかかる問い合わせを処理するため、比較に時間がかからない述語を順に適用し候補集合を絞り込むというフィルタリング方法が有効であることが確認された[1]。この方法ではいままでは論理積（conjunction）による絞り込み効率だけが議論されたが、本文では論理和(disjunction)の必要性を主張し、従来のモデルに論理和による方法を加えて、拡張したジェネラール・フィルタリングを提案する。さらにコストモデルを樹立し、比較結果に基づいて有効性を検証する。

2　モティベーション

近年注目を集めている空間オブジェクト（二次元平面上の多角形か複数の曲線線分で囲まれる図形）の集まりを扱う空間データベースシステムにおいて、オーバーラップ（overlap）を述語とする結合演算や選択演算は、代表的な操作として知られている。ところが、空間オブジェクトのオーバーラップ判定のような演算は、計算量が非常に大きいという問題を抱えている。このような問題には、フィルタリング方法が有効である。

図1では4つの空間オブジェクトo_1，o_2，o_3，o_4とそれぞれの最小包囲矩形（Minimum Bounding Rectangle: MBR）を示す。この中に、本当にオブジェクトo_1とオーバーラップしたのはo_4だけである。実際に、o_2，o_3，o_4とo_1とのオーバーラップ関係を調べる前にフィルターをかける方がいい。例えば、オブジェクトo_3とo_1の最小包囲矩形がオーバーラップしないので、オブジェクト自身のオーバーラップがあり得ない。2つの最小包囲矩形に対するオーバーラップはそれぞれの矩形の2点の座標の位置だけで判定できるので、計算量は、空間オブジェクトに対するオーバーラップと比較して非常に少ない。また、最小包囲矩形がオーバーラップしてもオブジェクト自身がオーバーラップしない(例:o_1とo_2)ことを判定する述語もさまざまあり、こういう述語も用いれば、o_1とのオーバーラップを判定しなければならないのはo_4だけで、対象数が最初の3から1に減らすことができる。

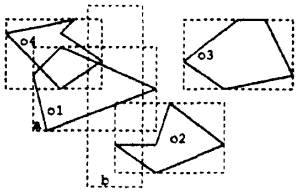

図 1: オーバーラップの例

二つの任意の空間オブジェクトx，yに対するオーバーラップを$overlap(x,y)$と書き、2つの任意の空間オブジェクトx，yの最小包囲矩形に対するオーバーラップを$MBRovlap(x,y)$と書くと次のようになる。

$$overlap(x,y) \leftrightarrow MBRovlap(x,y) \wedge overlap(x,y) \quad (1)$$

さらに、図の例からみると、オブジェクトo_1の最小包囲矩形がaであり、aと十字交差するbを最小包囲矩形とするオブジェクトはo_1とオーバーラップしなければならない。明らかにオーバーラップするオブジェクトに対して、述語$overlap$を適用しなくてもいい。

このような最小包囲矩形の位置関係を$MBRcross(x,y)$と書くと、(1)式は次のようになる。

$$overlap(x,y) \leftrightarrow MBRcross(x,y) \vee (MBRovlap(x,y) \wedge overlap(x,y)) \quad (2)$$

このようにして、$MBRcross(x,y)$や$MBRovlap$などをフィルタとして用い、明らかにオーバーラップし

Evaluating Expensive Queries, From Index to General Filtering

Hanxiong CHEN†, Nobuo Ohbo‡

†Department of Industrial Information, Tsukuba Inte'l Univ.

‡Institute of Information Sciences and Electronics, Univ of Tsukuba

ないもしくはオーバーラップするというオブジェクトを除外したら $overlap$ の評価対象が減る。如何にフィルタの適用順番を決め、全体の計算コストを効率良く減らすことが本研究の目的である。

3 ジェネラル・フィルタリング

一般的に、オブジェクトの集合 R から関係 f を満たす要素を求める操作を $\sigma_f R$ で表すが、本文では混乱が起こらない限り、簡単に f と書く。$\sigma_{f_n}(...(\sigma_{f_2}(\sigma_{f_1}R)))$ をリスト $(f_1, f_2, ..., f_n)$ で表す。

計算量の多い述語 f に対して、述語 f_1, f_2, ..., f_n がそれぞれ f の必要条件となっていれば、次のように書ける。

$$f \rightarrow f_1 \wedge f_2 \wedge ... \wedge f_n \qquad (3)$$

(3) 式の左辺と右辺が等価であるためには、$f_1, f_2, ..., f_n$ のいずれかが f 自身であるか、または f の必要十分条件となっていなければならない。f_1, f_2, ..., f_n の中、計算量の少ない述語は f のフィルターになりえ、このようなフィルターを用いることで f にかけられる計算対象を大幅に減らすことができる。

f_1, f_2, ..., f_n の実行順序は、フィルターとしての効果と計算量を考慮して決められる [1]。

また、述語 g_1, g_2, ..., g_n がそれぞれ f の十分条件となっていれば、次のように書ける。

$$f \leftarrow g_1 \vee g_2 \vee ... \vee g_n \qquad (4)$$

以上の一般形式は

$$f \leftrightarrow f_1 \ op \ f_2 \ op \ ... \ op \ f_n \qquad (5)$$

で op が $\wedge$ か $\vee$ のいずれであるが、その性質は非常に複雑である。今回は各 f_{ij} が互いに独立の仮定で右辺が

$$\wedge_i (\vee_j f_{ij}) \qquad (6)$$

の問い合わせに対して議論を行なう。

明らかに、次の式が成り立つ。

$$\sigma_{\vee_i f_i} R = \cup_i \sigma_{f_i} R = R - \sigma_{\wedge_i \neg f_i} R$$

しかし、実際の問い合わせ評価に際して、右辺の方が効率がいい。従って、まず (6) を変形し、

$$\wedge_i (\neg(\wedge_j \neg f_{ij})) \qquad (7)$$

が得られる。便宜上、$\wedge_j \neg f_{ij}$ を g_i とおき、f_{ij}, g_i のセレクティビティ (selectivity) と計算コストをそれぞれ下表のようにすると、[1] の結論がここでも成り立つ。

	g_i	f_{ij}
selectivity	s_i	s_{ij}
cost	λ_i	λ_{ij}

Lemma. 述語 f_p のフォルスドロップ (false drop)d_p とランク r_p をそれぞれ

$r_p = \lambda_p/(1 - d(p))$、$d_p = s_p/(1 - s_p)$ としたら、

$Cost((f_1, ..., f_n)) = min\{Cost((f_{i_1}, ..., f_{i_n}))\}$

以上の結論により、(7) に対して、次の最適評価手順が得られる。

1. g_i のランクの昇順に従い、各 $(\neg(\wedge_j \neg f_{ij}))$ を並べる。
2. それぞれの $(\wedge_j \neg f_{ij})$ に対して、f_{ij} のランクの降順に従い f_{ij} を並べる。

4 おわりに

本研究では、計算量の大きい述語に対して絞り込み効果のある (フィルター効果のある) 述語について調べた。今後は、f_{ij} の独立性制限を外し、(5) の一般形に対するより系統的なフィルターの検討、個々の述語のコスト評価、及びスクリーニング効果の見積もりを行なうと共に、最適な述語の実行順序及び組み合わせについて検討する。

参考文献

[1] H. Chen, *et al.* Decomposition - An approach for optimizing queries including ADT functions. *Information Processing Letters.* Vol.43, pp.327-333, 1992

[2] Dimitris Papadias, Yannis Theodoridis, Timos Sellis, Max J. Egenhofer. Topological Relations in the World of Minimum Bounding Rectangles: A Study with R-trees. *ACM SIGMOD*, pp.92-103, 1995