情報容量を考慮したSuperSQLクエリ作成支援

大澤 幸子† 遠山 元道†

† 慶應義塾大学 理工学部 情報工学科

1 はじめに

SuperSQL[1, 2] は同じ属性集合に対するクエリ指定においても、様々な表示レイアウトの指定ができる。しかし、レイアウトによっては結果表示においてクエリの対象となるデータベース上の属性間の関連が正しく反映されず、情報が適切に表現されないことがある。そこで、本研究では情報容量概念を定義し、データベース*の情報容量およびクエリの情報容量を比較することによって情報の損失の有無を判定し、判定結果をユーザに提示することで、クエリ作成を支援するシステムを提案する。

2 SuperSQL

SuperSQL のクエリ文は SQL の SELECT 句を GENERATE `<medium>` `<TFE>` の構文をもつ GENERATE 句に置き換えたものである。ここで`<medium>` は出力媒体の指定で、HTML, Java, Excel, LaTeX などの指定が可能である。`<TFE>` はターゲットリストの拡張である Target From Expression を表し、結合子、反復子などのレイアウト指定演算子をもつ一種の式である。

2.1 結果表示の状態の例

情報が適切に表現されない状態について、具体的に例を挙げて説明する。表 1 に表した 3 項関係に対し、次のような LaTex の表を生成する SuperSQL クエリ文

GENERATE latex
[都道府県, [駅名]!, [鉄道会社]!]!
FROM T

を実行すると、結果は表 2 のようになる。

表 2 を見てわかる通り、それぞれの都道府県に属する駅、鉄道会社を独立に都道府県名によってグルーピングしているため、この結果表示においてはどの駅にどの鉄道が通っているかという情報が表現されていない。

表 1: 3 項関係 S

都道府県	駅名	鉄道会社
東京都	渋谷	都営地下鉄
東京都	渋谷	東京急行
東京都	品川	都営地下鉄
東京都	品川	京浜急行
神奈川県	横浜	東京急行
神奈川県	横浜	京浜急行

表 2: クエリの結果

東京都	渋谷 品川	都営地下鉄 東京急行 京浜急行
神奈川県	横浜	東京急行 京浜急行

このようにもとのデータベースに含まれる情報が、表示結果に置いて表現されない、また逆に、もとのデータベースには含まれない情報があたかも存在するように表示結果に現れる、といった状態について情報容量を定義することで体系付けていく。

3 情報容量

定義 3.1 (情報容量)
クエリおよびその対象となるデータベースが含む属性集合において、それぞれの属性間の関連の有無によって表現可能な情報の大小に関する半順序関係を、情報容量とする

定義 3.2 (情報容量式)
情報容量を和積の形で表現した式を情報容量式とする *(A,B は任意の積節を表す)*

1. 属性間の関連を直接的に表現できる場合、それらの属性の積項として式に表す
   但し、 $A * B = B * A$
   $A * A * B = A * B$
2. 属性の積項で表される情報容量を複数含む場合、それらの積項の和として式にあらわす

Support for SuperSQL querying with concidering Information Capacity
OSAWA Sachiko†, TOYAMA Motomichi†
†Department of Information and Computer Science, Faculty of Science and Technology, Keio University.

*但し、データベースは第 5 正規形であるとする

但し、 $A + B = B + A$

$A + A * B = A * B$

例 3.1 属性 *A, B, C* を含む関係 *R(a, b, c)* の情報容量は $\phi(R) = a * b * c$

例 3.2 関係 *R(a, b, c), S(d, e, f)* を結合した情報容量は $\phi(R \bowtie S) = a * b * c + d * e * f$

定義 3.3 (結果表示の状態)
データベースの情報容量 $(\phi(DB))$ とクエリの情報容量 $(\phi(Q))$ を比較し、それらの大小関係により、結果表示の状態をそれぞれ以下のように定義する

1. $\phi(DB) = \phi(Q)$ の場合を**適正表示**とする
2. $\phi(DB) \neq \phi(Q)$ の場合を**表示異常**とする
   (a) $\phi(DB) > \phi(Q)$ の場合を**過少表示**とする
   (b) $\phi(DB) < \phi(Q)$ の場合を**過大表示**とする

4 システム概要

4.1 情報容量の算出

情報容量は、クエリ文の TFE 部より算出する。

4.1.1 クエリの情報容量

TFE にはそれぞれの属性に対するレイアウト指定演算子や装飾子が含まれているが、レイアウトの方向や装飾がいかなるものであれ、属性間の関連に影響することはない。つまりクエリの情報容量を算出する際には、TFE のネスト構造にのみ着目する。TEF より抜き出したネスト構造に、以下の規則を適応し、クエリの情報容量を求める。

Rule 1 $\phi(a) = a$
Rule 2 $\phi([A]) = \phi(A)$
Rule 3 $\phi(aA) = a * \phi(A)$
Rule 4 $\phi([A][B]) = \phi([A]) + \phi([B])$
Rule 5 $\phi(a[A][B]) = a * \phi([A]) + a * \phi([B])$
(a は任意の属性、A,B は任意の TFE を表す)

4.1.2 データベースの情報容量

データベースの情報容量を求めるには、さらにクエリ文の WHERE 句に含まれる属性にも着目する。ここで着目したいのは属性間の関連であるので、WHERE 句を構成する条件文の内、ある属性に対して定数や価の範囲を指定するものは除外し、両辺が供に属性であるもののみを考慮する。以下にデータベースの情報容量を求める手順をまとめる。

1. TFE に含まれる属性に着目
   同一テーブルの属性を積項にまとめ、それぞれのテーブルの積項を和算する
2. WHERE 句に含まれる属性に着目
   条件文の左辺または右辺の属性に
   (a) テーブルの主キーがある場合
   条件文で結ばれる 2 つのテーブルの項 (1 で求めたもの) を積算し、1 つの積項にまとめる
   (b) TFE に含まれる属性がある場合
   その属性を、条件文の他辺の属性のテーブルの項 (1 で求めたもの) に積として加える

4.2 情報容量の比較

情報容量の比較は、クエリ、データベースそれぞれの情報容量式に含まれる積項の単位で行う。以下の規則を積項の全ての組み合わせに適応し、それらの大小を判定する。

$A < A * B$ (A,B は任意の積節を表す)

それぞれの情報容量式を以下のように表し、

$$\phi(Q) = Q_1 + Q_2 + \cdots + Q_n$$
$$\phi(DB) = DB_1 + DB_2 + \cdots + DB_m$$

$0 \leq i \leq n,\ 0 \leq j \leq n$ とすると、

1. ${}^{\forall i}{}^{\exists j}\ (Q_i = DB_j)$ のとき適正表示
2. ${}^{\exists j}{}^{\forall i}\ (Q_i < DB_j)$ のとき過少表示
3. ${}^{\exists j}{}^{\forall i}\ (Q_i > DB_j)$ のとき過大表示
4. それ以外のとき過少表示と過大表示の混合状態

5 まとめ

本システムにより、SuperSQL クエリのレイアウト指定による表示異常の発生が検出可能となった。今後はさらに、レイアウト指定の改善方法もユーザに提示できるようなツールとして発展させることを検討している。

情報容量の論理を XML の情報流通や、情報統合における情報の損失や矛盾の検出に応用させることも検討している。

参考文献

[1] M. Toyama, SuperSQL: An Extended SQL for Database Publishing and Presentation, in *Proc. SIGMOD '98*, ACM(1998), pp.584-586.

[2] SuperSQL, http://www.db.ics.keio.ac.jp/ssql

[3] 遠山 元道 : 『データベース出版における木構造スキーマの情報容量』、情処研報、Vol.98,No.58、P173-180、情報処理学会、1998