データベースシステム 120－7
（2000．1．24）

不完全多次元データベースにおける集約可能性

石井拓　上林彌彦
京都大学 情報学研究科 社会情報学専攻
{hiromu, yahiko}@isse.kuis.kyoto-u.ac.jp

データベースによる意思志決定支援の有力な手法の一つに OLAP がある．しかし，データがある種の条件を満足していなければ，集約計算の結果として得られる解が正しくない，すなわち集約計算が可能でないことがあり，現状の OLAP ではそのような場合を扱うことが困難である．
本研究では広く普及した OLAP ツールである多次元データベースにおいて集約可能性問題を定式化する．この定式化は多次元データベースが不完全な情報，具体的には未知の値を扱うことを許しているという点で先行研究の拡張になっている．さらに集約計算が正しい解をもたらすための必要十分条件を求める．

Summarizability in Incomplete Multidimensional Databases

Hiromu Ishii　Yahiko Kambayashi
Department of Social Informatics, Graduate School of Informatics, Kyoto University

In a multidimensional database(MDDB), a popular tool for OLAP, correct summarization of data is extremely important for data analysis and decision support activities. However, without considering summarizability conditions, summarization over multidimensional databases yields wrong and erroneous results which could affect business decisions.
In this paper, firstly we formalize the summarizability problem in an MDDB which may be incomplete, i.e. is allowed to have "unknowns". Then we prove the necessary and sufficient condition for summarizability.

1　はじめに

近年大量データに基づいて意思決定を行うためのデータベース技術が注目を浴びているが，その中の一つに **OLAP(On-Line Analytical Processing)** がある．OLAP は利用者に様々な視点から対話的にデータを分析することを可能にさせるもので，総和などの集約計算 (summarization) を含む問合せを多用することを特徴の一つとする．以下は商品毎日毎地域毎の売上額を格納したデータベースに対する OLAP 問合せの例である．

例 1　OLAP 問合せ

- 市町村毎の総売上額は？
- 月毎の一日平均売上額は？
- 総売上額で上位から 5 番目までの製品は？

OLAP を実現するツールとしては，その直観的な分かりやすさから**多次元データベース (MDDB : Multidimensional Databases)** が現在の主流となっている [Ram98]．MDDB は分析の対象である事象 (以後単に事象) について利用者が着目している量を，事象の属性を各次元とする多次元の立方体上に配置されたものとして見せる，一種のフロントエンドツールである．図 1は売上 (商品 ID, 時刻 ID, 地域 ID, 金額) リレーション (左) とそれを 3 次元の MDDB に変換したもの (右) の対応を示している．

誤った意思決定を防ぐ上で集約計算の結果の正確さは非常に重要であるが，米国立バークレー研究所の Shoshani らによって MDDB における集約計算が必ずしも正しい解，言い換えれば利用者の意図した解を返さないことが指摘されている [RS90, LS97]．以下は彼らの論文からの引用である．

pid	tid	locid	sales
11	1	1	8
11	2	1	25
.	.	.	.
.	.	.	.
.	.	.	.

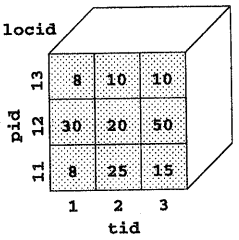

図 1: リレーションと多次元データベース

例 2 集約計算における問題

下はある大学の工学部の学科毎 (2 学科しかないとする) 年度毎の在籍学生数を格納した 2 次元 (「大学の組織」と「時間」) の MDDB である.

$DptYear\_\#Std$		
$Dpt\backslash Year$	1994	1995
CS	15	17
Statistics	10	15

この MDDB に対して工学部全体の年度毎の在籍者を求めたい場合, 2 学科の数字を足して

$Year\_\#Std(1)$		
$Dpt\backslash Year$	1994	1995
total	25	32

とすることはごく自然である. しかし, もし工学部が学生に 2 つの学科に同時に所属することを許しており, かつ各年度 2 人ずつその制度を利用している学生がいたならば利用者の意図を反映した解は, 上の結果の各数字から 2 を引いた次の $Year\_\#Std(2)$ であるべきである.

$Year\_\#Std(2)$		
$Dpt\backslash Year$	1994	1995
total	23	30

すなわち MDDB $DptYear\_\#Std$ は次元「大学の組織」における総和計算に関し集約可能でないと言える. (例終)

Shoshani らは他にもいくつか集約計算が正しい解を返さない例を挙げ, それらの分析をもとに集約可能であるための十分条件を述べている. が, 彼らの議論は STORM[RS90] という統計データベース用の一種の抽象モデル上で非形式的に展開されており, 実際の MDDB に適用可能なレベルまでには至っていない. そこで筆者らは, Snowflake モデル [Kim96] と呼ばれる現実に用いられている MDDB のモデル上で集約可能性問題の定式化および集約可能となる必要十分条件 (以下集約可能条件) の提示を行った [石井 99].

しかし, MDDB のモデル研究の進展に伴い, 最近では Snowflake モデルは実用的な MDDB モデルとしては不十分であるとの指摘がされるようになっている [PJ99]. 具体的には Snowflake モデルには「次元に階層関係が存在する場合, インスタンス間の上下 (包含, 所属) 関係の記述は隣接する階層間でしか出来ない」という制約があるが (例えば「大学の組織」次元が存在し, そこに "「学生」<「学科」<「学部」" という階層構造が存在する場合, 学生インスタンスと学科インスタンスの間の所属関係, および学科インスタンスと学部インスタンスの間の所属関係は記述できても, 学生インスタンスと学部インスタンスの間の所属関係を直接記述することは出来ない), 現実の応用では隣接しない階層に属するインスタンス間の関係も記述可能であってほしい場合が多く見られる, といった点である.

そこで本研究では, Snowflake モデルよりも現実的な MDDB モデルを採り上げ, あらためて集約可能性問題に取組んだ. より具体的には, 上記のような制約のない, すなわち "学生 A は理学部に所属することは判明しているが, 所属学科は分からない" といった不完全な情報も扱えるような MDDB モデルを用いて, その上で集約可能性問題を定式化し, さらに集約可能条件を求めた.

2 基本事項

2.1 駐車違反 MDDB

本稿では以降を通じて「大学における駐車違反 MDDB」(以下 $PVDB$[1]) を例として用いる. $PVDB$ は「駐車違反の記録 ($Record$)」を事象とし, 分析に用いる次元として「大学の組織 (Org)」, 「時間 ($Time$)」, 「回数 ($\#V$)」を備えている. さらに, はじめの 2 つの次元は各々 "学生 (Std)< 学科 (Dpt)< 学部 (Scl)", "学期 (Sem) < 年度 ($Year$)" という階層構造を持っている.

図 2は $PVDB$ のあるインスタンスを図示したものである. たとえば事象 $r01$ から $std01, 1998F, 4$ に矢印が出ていることは, $r01$ が「学生 $std01$ は 1998 年の秋学期 ($1998F$) に 4 回駐車違反を犯した」という記録であるこ

[1] PV=Parking Violation

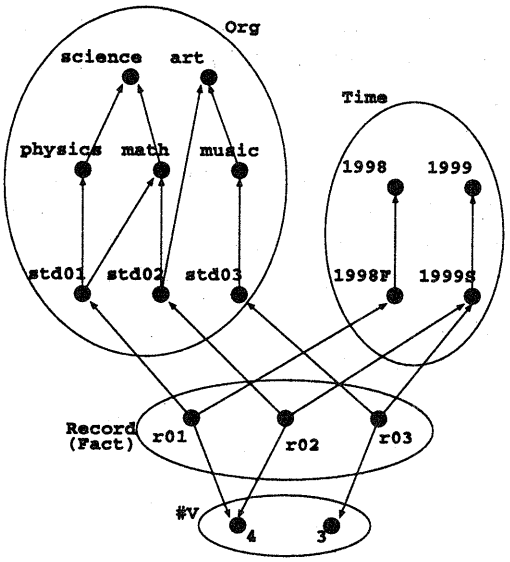

図 2: 駐車違反 MDDB($PVDB$) のインスタンス

とを表している.

2.2 MDDB のデータモデル

MDDB のモデルとしてはこれまで様々なものが提唱されているが, 本研究では OLAP における MDDB への要求を我々の知るかぎり現時点で最もよく満たしていると思われる Pedersen らのモデル [PJ99] を採用する.

• 事象スキーマ

n 次元の事象スキーマ S を 2- タプル (F, D) で定義する. ここで F は事象の型で, $\mathsf{D} = \{\mathsf{T}_i, (i = 1, \ldots, n)\}$ は F の元を特徴づけるのに用いられる n 個の次元の型の集合である.

例 3 $PVDB$ においては Record が事象の型で, $\{\mathsf{T}_1, \mathsf{T}_2, \mathsf{T}_3\}$ = {Org, Time, #V } が次元型集合である.

• 次元型, カテゴリ型

次元型 T は 4- タプル $(\mathsf{C}, \leq_\mathsf{T}, \top_\mathsf{T}, \bot_\mathsf{T})$ として定義される. $\mathsf{C} = \{\mathsf{C}_j\}(j = 1, \ldots, k)$ は型 T の次元に含まれる (k 個の) カテゴリの型の集合, $\leq_\mathsf{T}$ は $\top_\mathsf{T}, \bot_\mathsf{T}$ をそれぞれ最大元, 最小元とする C 上の半順序である (すなわちある次元型に対応するカテゴリ型の集合は束をなす). 直観的には, カテゴリ型 C_1 の外延の集合がカテゴリ型 C_2 の外延の集合を含むとき, $\mathsf{C}_2 \leq_\mathsf{T} \mathsf{C}_1$ (C_1 は C_2 より大きい, あるいは C_1 は束上 C_2 より上にある) となる.

例 4 $PVDB$ において, 型が Org の次元中のカテゴリ型は $\bot_{\mathsf{Org}} = \mathsf{Std} < \mathsf{Dpt} < \mathsf{Scl} < \top_{\mathsf{Org}}$ である.

• 次元, カテゴリ

型が $\mathsf{T} = \{\{\mathsf{C}_j\}, \leq_\mathsf{T}, \top_\mathsf{T}, \bot_\mathsf{T})$ である次元 D は 2- タプル $D = (C, \leq)$ で定義される. ここで $C = \{C_j\}$ は $Type(C_j) = \mathsf{C}_j$[2]であるようなカテゴリの集合で, 型が C_j であるカテゴリ C_j は $Type(e) = \mathsf{C}_j$ であるような値 (次元値と呼ぶ) e の集合である. また, $\leq$ は $\cup_j C_j$ つまり D 中の全カテゴリの次元値すべてを含む集合における半順序で, $\leq$ はもし二つの次元値 e_1, e_2 について, e_1 が論理的に e_2 に含まれる (「e_1 ならば e_2」が成り立つ) ならば $e_1 \leq e_2$ で定義される. なお, $C_j \in C$ のとき C_j は D のカテゴリであるといい, そのことを $C_j \in D$ で表す. また $e \in \cup_j C_j$ のとき e は D の次元値であるといい, $e \in D$ で表す.

次元型 T の カテゴリ型 $\bot_\mathsf{T}$ は最も小さいサイズ, すなわちどんな次元値も真には含まれない次元値の型である. また, カテゴリ型が $\top_\mathsf{T}$ である次元値はただ一つ (T で表す) で, $\forall e \in D(e \leq \mathsf{T})$ である.

例 5 図 2の $PVDB$ インスタンスの次元 Org では次元値の間に次のような半順序が存在する.

$\{std01 < physics$, $std01 < math$, $std02 < math$, $std02 < art$, $std03 < music$, $physics < science$, $math < science$, $music < art\}$

このモデルでは $std02 < art$ のように順序 $\leq$(階層) で隣接していないカテゴリ間のインスタンスの順序関係も記述できるので, 「学生 $std02$ は芸術学部 (art) に所属しているが, 所属学科は分からない」といった不完全な情報が扱える.

• 事象 - 次元リレーション

F を事象集合, $D = (\{\mathsf{C}_j\}, \leq)$ を次元とする. F と D の間の事象 - 次元リレーション (fact-dimension relation) は集合 $R = \{(f, e)\}$ で定義される. ここで $f \in F$ かつ $e \in \cup_j C_j$. $\exists e_1 \in D((f, e_1) \in R \wedge e_1 \leq e)$ が成り立つとき**事象** f は**次元値** e で**特徴づけられる**と言い, $f \hookrightarrow e$ で表す. なお, 任意の事象は次元中のある次元値に結び付けられている, すなわち $\forall f \in F(\exists e \in \cup_j C_j((f, e) \in R))$ は常に成立しているとする (次元値が未知の場合には

[2] $Type$ は型を返す関数.

$(f, \top)$ を R に加える).

例 6　図 2の $PVDB$ のインスタンスにおいては，事象 $Record$ と次元 $Time$ の間の事象 - 次元リレーションは $\{(r01, 1998F), (r02, 1999S), (r03, 1999S)\}$ となる.

- **多次元オブジェクト**

多次元オブジェクト (MO : Multidimensional Object) を 4- タプル $M = (\mathsf{S}, F, D, R)$ で定義する．ここで，$\mathsf{S} = (\mathsf{F}, \mathsf{D} = \{\mathsf{T}_i\})$ は事象スキーマ，$F = \{f\}$ は $Type(f) = \mathsf{F}$ である事象 f の集合，$D = \{D_i\}(i = 1, \ldots, n)$ は $Type(D_i) = \mathsf{T}_i$ である次元 D_i の集合，$R = \{R_i\}(i = 1, \ldots, n)$ は $\forall i((f, e) \in R_i \Rightarrow f \in F \wedge \exists C_j \in D_i(e \in C_j))$ である事象 - 次元リレーションの集合である.

- **多次元データベース**

多次元データベース (MDDB) は多次元オブジェクトの集合である.

2.3　集約計算

Pedersen らは 2.2 で述べたデータモデルと共にその上のデータ操作言語 (代数) も提唱している [PJ99]．代数の構成要素である演算のうち，ここでは本研究に特に関連のある集約演算の定義のみを述べる．なお，集約可能性の議論を行うために独自の変更を加えてある.

先立って補助的な関数 $Group$ の定義を行う.

· $Group$

[入力]

n 次元 MO $M = (\mathsf{S}, F, \{D_i\}, \{R_i\})$，$M$ の各次元から一つずつ取った n 個のカテゴリを要素とする集合 $C = \{C_i \mid C_i \in D_i\}$，および n- タプル $(e_1, \ldots, e_n)(e_i \in C_i)$

[出力]

$Group(e_1, \ldots, e_n)$[3] $= \{f \mid f \in F \wedge f \hookrightarrow_1 e_1 \wedge \cdots \wedge f \hookrightarrow_n e_n\}$

$Group$ は同じ次元値の組合わせによって特徴づけられる事象をグループ化する関数である.

- **集約演算**

集約演算 (記号 α) を以下で定義する.

[入力]

n 次元の MO M，新しく作られる次元 D_{n+1}，事象の集合から[4] D_{n+1} の次元値への関数 $g \in \{$SUM,CND[5], AVG, MAX, MIN$\}$，何番目の次元に g を施すかを指定する数字 i，カテゴリ集合 $\{C_1, \ldots, C_n\}(C_i \in D_i)$

[出力]

$\alpha[D_{n+1}, g(i), C_1, \ldots, C_n](M) = (\mathsf{S}', F', D', R')$

すなわち結果は MO となる．ここで g が CND の場合には前処理として M の構成要素を以下の手順で変更する必要があるが (この部分が我々が加えた変更である)，

1. 事象 - 次元リレーション集合 $R_1'', \ldots, R_n''$ を以下のようにして作る.
   - $(f, d) \in R_j$ かつ $f \hookrightarrow e(e \in C_i)$ ならば $(e, d) \in R_j''$ $(j \neq i)$
   - $e \in C_i$，$d \in D_i$ について，$e \leq d$ ならば $(e, d) \in R_i''$
2. R_k を $R_k''(k = 1, \ldots, n)$ で置き換える.
3. F を C_i で置き換える.
4. F を C_i で置き換える.

結果 MO の各構成要素は以下で定義される.

- $\mathsf{S}' = (\mathsf{F}', \mathsf{D}')$，
  $\mathsf{F}' = 2^{\mathsf{F}}, D' = \{\mathsf{T}_i', i = 1, \ldots, n\} \cup \{\mathsf{T}_{n+1}\}, \mathsf{T}_i' = (\mathsf{C}_i', \leq_{T_i}', \perp_{\mathsf{T}_i}', \top_{\mathsf{T}_i}'), \mathsf{C}_i' = \{\mathsf{C}_{ij} \in \mathsf{T}_i \mid Type(C_i) \leq_{\mathsf{T}} \mathsf{C}_{ij}\}, \leq_{\mathsf{T}_i}' = \leq_{\mathsf{T}_i|_{\mathsf{C}_i'}}, \perp_{\mathsf{T}_i}' = Type(C_i), \top_{\mathsf{T}_i}' = \top_{\mathsf{T}_i}$,
- $F' = \{Group(e_1, \ldots, e_n) \mid (e_1, \ldots, e_n) \in C_1 \times \cdots \times C_n \wedge Group(e_1, \ldots, e_n) \neq \emptyset\}$,
- $D' = \{D_i', i = 1, \ldots, n\} \cup \{D_{n+1}\}$，
  $D_i' = (C_i', \leq_i'), C_i' = \{C_{ij}' \in D_i \mid Type(C_{ij}') \in \mathsf{C}_i'\}, \leq_i' = \leq_{i|_{D_i'}}$,
- $R' = \{R_i', i = 1, \ldots, n\} \cup \{R_{n+1}'\}$，
  $R_i' = \{(f', e_i') \mid \exists (e_1, \ldots, e_n) \in C_1 \times \cdots \times C_n (f' = Group(e_1, \ldots, e_n) \wedge f' \in F' \wedge e_i = e_i')\}$,
  $R_{n+1}' = \cup_{(e_1, \ldots, e_n) \in C_1 \times \cdots \times C_n} \{(Group(e_1, \ldots, e_n), g(Group(e_1, \ldots, e_n))) \mid Group(e_1, \ldots, e_n) \neq \emptyset\}$

例 7　図 2の $PVDB$ インスタンスを MO P とする．「学科毎の駐車違反回数の総和」を求める集約計算は

[3] M, C は誤解の恐れがないかぎり省略する.

[4] 事象から集約関数を施される次元の値へ lookup を行うと考える.

[5] COUNT DISTINCT の略．重複なしで数を数える関数.

$P' = \alpha[Vsum(= D_4), \mathrm{SUM}(1), Dpt, \top_{\mathrm{Time}}\top_{\#\mathrm{v}}](P) = (\mathrm{S}, F, D, R)$ となり，$R_1 = \{(\{r01\}, physics), (\{r01, r02\}, math), (\{r03\}, music)\}$, $R_4 = \{(\{r01\}, 4), (\{r01, r02\}, 8), (\{r03\}, 3)\}$ となる．R_1 と R_4 から，例えば数学科 $(math)$ に所属する学生の犯した総違反回数は8であることが見てとれる．

本稿ではMOに集約演算を施すことを集約計算と呼び，MO M' が M から集約計算で生成されることを $M \rightarrow_\alpha M'$ で表す．また，新しく作られる次元 D_{n+1} を集約次元，集約関数を施される次元 D_i を被集約次元と各々呼ぶ．さらに，集約計算の結果ではないMOを base-MO であるという．現実的には base-MO はもっとも細かいレベルでの情報を格納したMOである．

3 集約可能性

Lenz らは集約可能性 (summarizability) を以下のように定義している [LS97].

“あるマクロデータ M に集約計算 S を施す場合，結果がマイクロデータから求めたものと一致すれば，「S の M における集約可能性は成立する」と言う”

ここでマイクロデータおよびマクロデータは統計データベースの分野で古くから使われている用語だが，形式的な定義はなく，各々「利用者が集約計算を施そうとしている，各個人もしくはオブジェクトについてのデータ」，「集約計算の結果として得られるデータ」[LS97] 程度の説明でコミュニティの合意を得ているようである．

図3は Lenz らの定義を視覚化したものである．DS はマイクロデータから直接求める場合の集約問合せ，$S_1, \ldots, S_M$ はマイクロデータから M を生成するのに用いられた集約問合せの並びである．

我々は Pedersen らのモデルを拡張することにより，Lenz らの定義を定式化した．

3.1 被集約事象 (Summarized Fact)

集約計算は自然に「集約関数を施される着目量(measure) をその次元属性として持つ事象をグループ化すること」と見なせる．

たとえば，例7の集約計算では，被集約次元は「(駐車違反) 回数」であったが，この集約計算は事象集合「駐車違反記録 」の要素をそれを犯した学生の学部毎に集めていることと等価である．この「駐車違反記録」を一般化した**被集約事象 (SF: Summarized Fact)** という概念を Pedersen らのモデルに導入する．

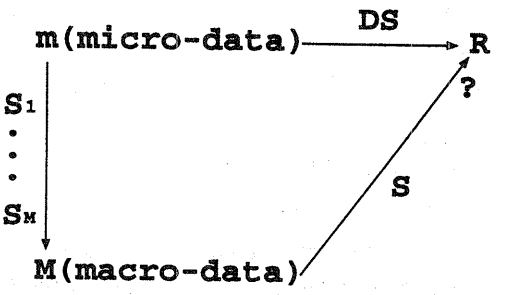

図 3: 集約可能性の概念図

- **被集約事象型**

  ある MO $M' = ((\mathrm{F}', \mathrm{D}'), F', D', R'))$ の被集約事象型 (SFT : Summarized Fact Type) $SFT(M')$ は以下のように定義される．

  - M' が base-MO ならば，$SFT(M') = \mathrm{F}'$
  - さもなければ，$M = ((\mathrm{F}, \mathrm{D}), F, D, R)$, $M' = \alpha[D_{n+1}, g(i), C_1, \ldots, C_n](M)$ であるとして
    - g が CND であるか，AVG であるか，あるいは D_i が M の集約次元でなければ，$SFT(M') = \mathrm{F}'$
    - g が MAX ならば，
      - もし $SFT(M) = \mathrm{T}_{\mathrm{MAX}}$[6] かつ D_i が M の集約次元であれば，$SFT(M') = SFT(M)$
      - さもなければ $\mathrm{F}'_{\mathrm{MAX}}$
    - g が MIN ならば，
      - もし $SFT(M) = \mathrm{T}_{\mathrm{MIN}}$ かつ D_i が M の集約次元であれば，$SFT(M') = SFT(M)$
      - さもなければ $\mathrm{F}'_{\mathrm{MIN}}$
    - さもなければ，$SFT(M') = SFT(M)$

- **被集約事象集合，被集約事象**

  まず補助関数 $Flat$ を以下で定義する．

[6] 型 T に「最大値をとる」こと情報として加えた型．

·$Flat$

[入力]

集合 $S = \{s_i\}$, 型 T

[出力]

- もし $Type(S) = \mathsf{T}$ ならば, S.
- さもなければ, $Flat(\cup s_i, \mathsf{T})$.

ここで $\cup$ は集合和で, $Flat$ の出力は集合となる.

$Flat$ を用いて MO $M = ((\mathsf{F}, \mathsf{D}), F, D, R)$ の被集約事象集合 (SFS: Summarized Fact Set)$SFS(M)$ を以下のように定義する. なお, T に MAX もしくは MIN の添字がある場合には無視するものとする.

$$SFS(M) = Flat(F, SFT(M))$$

$SFS(M)$ の個々の要素を M の被集約事象と呼ぶ.

例 8 例 7 の P' において,

$SFS(P') = Flat(\{\{r01, r02\}, \{r01\}, \{r03\}\}, 2^{\mathsf{Record}})$
$= \{r01, r02, r03\}$

- **被集約事象分布**

次に導入する被集約事象分布 (SFD: Summarized Fact Distribution) という概念は, 直観的には集約計算の結果 MO $M' = \alpha[D_{n+1}, g(i), C_1, \ldots, C_n](M)$ を, $C_1, \ldots C_n$ を各次元とする座標系と見て, 座標 $(e_1, \ldots, e_n)$ に $Group(e_1, \ldots, e_n)$ を配置したものと考えられる. SFD の定義に先立って $Flat$ を multiset(重複を許す集合) を扱うように変更した補助関数 $mFlat$ を定義する.

·$mFlat$

[入力]

要素の重複を許す集合 (multiset)$S = \{s_i\}$, 型 T

[出力]

- もし $Type(S) = \mathsf{T}$ ならば, S.
- さもなければ, $mFlat(\uplus s_i, \mathsf{T})$.

ここで $\uplus$ は重複を許す集合和である. すなわち $mFlat$ の出力は multiset となる.

$mFlat$ を用いて $M = \alpha[D_{n+1}, g(i), C_1, \ldots, C_n](M')$ の被集約事象分布 $SFD(M)$ を以下で定義する,

$SFD(M) = \{(mFlat(\{f\}, SFT(M)), e_1, \ldots, e_n)\}$

ここで $e_i \in C_i$, $Group(e_1, \ldots, e_n) \neq \emptyset$, $\{f\} = Group[M, C](e_1, \ldots, e_n)$ である.

例 9 例 7 の P' について

g''	g	g'
CND	SUM	CND
SUM	SUM	SUM
MAX	MAX	MAX
MIN	MIN	MIN

図 4: g' の生成テーブル

$SFD(P') = \{(\{r01, r02\}, math, \mathsf{T}_{\mathsf{Time}}, \mathsf{T}_{\#\mathsf{V}}),$
$(\{r01\}, physics, \mathsf{T}_{\mathsf{Time}}, \mathsf{T}_{\#\mathsf{V}}), (\{r03\}, music, \mathsf{T}_{\mathsf{Time}}, \mathsf{T}_{\#\mathsf{V}})\}$

3.2 直接集約計算

ここでは直接集約計算 (direct summarization) という概念を導入する. その前に前出のマイクロデータ, マクロデータに対応する概念を定式化する.

- **マクロ多次元オブジェクト**

集約計算の結果として生成された MO をマクロ多次元オブジェクト (macro-MO) と呼ぶ.

- **マイクロ多次元オブジェクト**

ある macro-MO M が base-MO M_0 から j 回の集約計算によって生成されているとする. すなわち $M_0 \to_\alpha M_1 \to_\alpha \cdots \to_\alpha M_j (= M)$ である. MO M_i が以下を満たすとき, M_i は M のマイクロ多次元オブジェクト (micro-MO) であるといい, $micro(M) = M_i$ で表す.

- $SFT(M_{i+1}) = 2^{\mathsf{G}}$ もしくは $2^{\mathsf{G}}_{\mathrm{MAX}}$ もしくは $2^{\mathsf{G}}_{\mathrm{MIN}}$ (G は M_i の事象型もしくはあるカテゴリ型)
- $SFT(M_{i+1}) = \cdots = SFT(M_j)$

- **直接集約計算結果**

$M' = \alpha[D_{n+1}, g(i), C_1, \ldots, C_n](M)$ のとき M' の直接集約計算結果 $DS(M')$ は以下で定義される.

$$DS(M') = \alpha[D_{n+1}, g'(i), C_1, \ldots, C_n](micro(M'))$$

ここで g' は M の直接計算結果が $DS(M) = \alpha[D_x, g''(y), C_z](M_p)$ であるとして, 図 4 のルールで決定される.

例 10 例 7 の P' を用いて「学部毎の総違反回数」を求めた macro-MO $P'' = \alpha[Vsum2(= D_4), \mathrm{SUM}(1), Scl, \mathsf{T}_{\mathsf{Time}}\mathsf{T}_{\#\mathsf{V}}](P')$ を考える. このとき $DS(P'') = \alpha[Vsum2, \mathrm{SUM}(1), Scr, \mathsf{T}_{\mathsf{Time}}\mathsf{T}_{\#\mathsf{V}}](P)$ である.

直観的には, macro-MO M の直接計算結果 $DS(M)$ は M によって利用者が得たい情報をもっとも詳細な情報源から求めたものと考えられる.

3.3 集約可能性

以上の準備を経て, 集約可能性を以下のように定義する.

"ある macro-MO M に対する集約計算 $\alpha(M)$ の結果を M' としたとき, $SFD(M') = SFD(DS(M'))$ のとき, またそのときにかぎり集約計算 $\alpha(M)$ は集約可能性を持つという."

例 11 例 10 の P'' について,

$SFD(P'') = \{(\{r01, r01, r02\}, science, \top_{Time}, \top_{\#V}), (\{r03\}, art, \top_{Time}, \top_{\#V})\}$

一方,

$SFD(DS(P'')) = \{(\{r01, r02\}, science, \top_{Time}, \top_{\#V}), (\{r02, r03\}, art, \top_{Time}, \top_{\#V})\}$

故に P'' を生成した集約計算は集約可能性を持たない.

4 集約可能となるための条件

4.1 分析

M をある macro-MO, $M' = DS(M)$ として, 3.3 で定義した集約可能性が成立しない状況を考える. ここで以下の 2 補題が成立することが容易に示せる.

補題 1

$SFD(M)$ のあるタプルが持つ次元値の組合せは $SFD(M')$ にも現れる. すなわち

$t \in \pi_{C_1,\ldots,C_n}(SFD(M))$

$\Rightarrow t \in \pi_{C_1,\ldots,C_n}(SFD(M'))$

補題 2

任意の $t \in \pi_{SFS}(SFD(M'))$ について, t は multiset だが, 要素の重複は存在しない.

上記 2 つの補題などから, $SFD(M) \neq SFD(M')$ となるのは次の各場合である (ただし排他的ではない).

[場合 1]

$\pi_{SFS}(SFD(M))$ が $\pi_{SFS}(SFD(M'))$ に真に含まれる ($SFD(M)$ と $SFD(M')$ のタプルの数が異なる).

[場合 2]

$\pi_{C_1,\ldots,C_n}(t) = \pi_{C_1,\ldots,C_n}(t')$ である $t \in SFD(M)$, $t' \in SFD(M')$ について, $\pi_{SFS}(t')$ が含むある被集約事象 $f \in SFS(M')$ を $\pi_{SFS}(t)$ が複数個含んでいる (例 11 の $SFD(P'')$ における $\pi_{Scl}((SFD(P'')) = (science)$ であるタプルが相当).

[場合 3]

$\pi_{C_1,\ldots,C_n}(t) = \pi_{C_1,\ldots,C_n}(t')$ である $t \in SFD(M)$, $t' \in SFD(M')$ について, $\pi_{SFS}(t')$ が含むある被集約事象 $f \in SFS(M')$ を $\pi_{SFS}(t)$ が含んでいない (例 11 の $SFD(P'')$ における $\pi_{Scr}((SFD(P'')) = (art)$ であるタプルが相当する).

4.2 集約可能条件

我々は上記 3 つの場合に対応する形で, 集約可能条件を作成した. あらためて

$M = \alpha[D_{n+1}, g(i), C = \{C_1, \ldots, C_n\}](M'')$, $M' = \alpha[D_{n+1}, g'(i), C_1, \ldots, C_n](micro(M))$, $M'' = (S'', F'', D'', R'')$ とする. また M'' には 4.1 に挙げたような 3 つの場合はいずれも生じていない. すなわち M'' は正しい集約結果を保持した macro-MO であるとする.

[条件 1]

$Group[micro(M), C](e_1, \ldots, e_n) \neq \emptyset$ である任意の次元値の組合せ $e_1, \ldots e_n$ について, $Group[M'', C](e_1, \ldots, e_n) \neq \emptyset$

[条件 2]

$Group[M'', C](e_1, \ldots, e_n) = \{f_i\}(\neq \emptyset)$ である任意の次元値の組合せ $e_1, \ldots e_n$ について, $Flat(f_k, SFT(M)) \cap Flat(f_l, SFT(M)) = \emptyset(f_k, f_l \in f_i, k \neq l)$

[条件 3]

$Group[M'', C](e_1, \ldots, e_n) = \{f_i\}(\neq \emptyset)$ である任意の次元値の組合せ $e_1, \ldots e_n$ について, $Group[micro(M), C](e_1, \ldots, e_n) \subseteq Flat(\{f_i\}, SFT(M))$

4.3 条件の必要十分性の証明

定理

4.1 の 3 条件が成立することが, 集約可能であるために必要十分である.

(十分性の証明)

対偶を示す.

〈[場合 1] が生じた場合〉

ある次元値の組合せ $e_1,\ldots,e_n$ について, $(e_1,\ldots,e_n) \in \pi_{C_1,\ldots,C_n}(SFD(M'))$ かつ $(e_1,\ldots,e_n) \notin \pi_{C_1,\ldots,C_n}(SFD(M))$ であるとする. SFD の定義から, ただちに $Group[micro(M),C](e_1,\ldots,e_n) \neq \emptyset$ かつ $Group[M'',C](e_1,\ldots,e_n) = \emptyset$. よって [条件 1] が非成立となる.

〈[場合 2] が生じた場合〉

ある $f \in SFS(M), t \in SFD(M)$ が存在して, $\pi_{SFS}(t) = \{\ldots,f,f,\ldots\}$ であるとする. M'' は正しい集約結果を保持しているとの仮定から, 任意の $f'' \in F''$ について, $mFlat(f'',SFT(M))$ は要素の重複を持たない. よって, ある $f_k, f_l \in SFS(M)$ が存在して, $f \in Flat(f_k,SFT(M))$ かつ $f \in Flat(f_l,SFT(M))$. よって $Flat(f_k,SFT(M)) \cap Flat(f_l,SFT(M)) \neq \emptyset$ で, これはすなわち [条件 2] の非成立を意味する.

〈[場合 3] が生じた場合〉

ある次元値の組合せ $e_1,\ldots,e_n$ について ある $f' \in Group\ [micro(M),C](e_1,\ldots,e_n)$ が存在して, 任意の $f \in Group\ [M'',C](e_1,\ldots,e_n)$ について $f' \notin Flat(f,SFT(M))$. これは [条件 3] の非成立を意味する.

(必要性の証明)

対偶が容易に示せることから略.

5 まとめと今後の方向

本稿では主要な OLAP ツールである多次元データベース (MDDB) において集約計算を行う際に不整合が生じる問題を論じた. MDDB のモデルとしては不完全な情報, 具体的には未知の値も扱えるモデル [PJ99] を採用し, その上で集約可能性問題を定式化した. さらに集約可能となる必要十分条件を提示した.

今後の方向としては当面以下を予定している.

- 集約可能条件の判定アルゴリズムの開発

  本研究での議論を実用に結び付けるには, より簡明な条件の作成から含めて, 効率的な条件判定アルゴリズムを設計する必要がある.

- 集約可能条件が成立しない場合の対処手法の開発

  集約可能性が非成立, すなわち直接の集約計算が成功しない場合に正しい解を求める手法を開発することは実用的に大きな意味がある。[PJD99] に見られる次元中あるいは事象 - 次元間の階層構造を解の正しさを保証しながら変形する手法はその一例である.

- 実装技術の開発

  本研究で採用したモデルは既存の他のモデルに比べて高い表現力を持つが, 抽象度もまた高い. 本 研究で導入した集約可能性を扱う機構を取り入れつつ実用的なシステムに結び付けるのは今後の課題である.

参考文献

[Kim96] R. Kimball "The Data Warehouse Toolkit" Wiley Computer Publishing, 1996.

[LS97] H. Lenz, A. Shoshani, "Summarizability in OLAP and Statistical Data Bases", In *Proc. of 9th International Conference on Scientific and Statistical Database Management(SSDBM)*, 1997.

[PJ99] T. B. Pedersen and C. S. Jensen "Multidimensional Data Modeling for Complex Data", In *Proc. of the 15th International Conference on Data Engineering(ICDE)*, 1999

[PJD99] T. B. Pedersen, C. S. Jensen and C. E. Dyreson "Extending Practical Pre-Aggregation in On-Line Analytical Processing" In *Proc. of the 25th International Conference on Very large Data Bases(VLDB)*, 1999

[Ram98] R. Ramakrishnan, "Database Management Systems" McGraw-Hill, 1998, ISBN 0-17-050775-9.

[RS90] M. Rafanelli, A. Shoshani , "STORM: A Statistical Object Representation Model", In *Proc. of 5th International Conference on Scientific and Statistical Database Management(SSDBM)*, 1990, pp. 14-29.

[石井 99] 石井拓, 上林彌彦, " 多次元データベースにおける集約可能条件", 情報処理学会アドバンスト・データベース・シンポジウム, pp. 161-170, 1999