

Kappaにおける集合とその内部処理

五十嵐 努* 金枝上 敦史** 河村 元大***

*三菱電機東部コンピュータシステム(株) **三菱電機(株)

*** (財)新世代コンピュータ技術開発機構

1. はじめに

知識ベース管理システムKappa(Knowledge Application oriented Advanced database management system)[1]は従来のデータベース管理システムに比べ、より自由なデータを表現するためにデータモデルとして非正規関係モデル(nested relational model)[1]を採用している。更にこのモデルを用いた非正規関係を操作する処理で集合という概念を導入している。集合は検索の結果として返され、集合間の演算(積、和、差、否定)を行うことでより複雑な検索を非正規関係を実際に読むことなく処理できる。集合は集合要素と呼ばれる選択情報の集まりで、集合要素はレコードの識別番号(RID:Record Identifier)と、繰り返しを持った属性の繰り返しを識別するサブレコード情報から成る。検索結果の読み出しは、集合要素とレコードの値とを対応付けることにより行う。

本論文では集合をどのような構造で実現しているのかについて述べた後、集合要素とレコードの値とを対応付ける処理の手順について述べる。

2. 集合の内部構造

集合は集合要素と呼ばれる選択情報の集まりで、集合要素はレコードの識別番号(RID:Record Identifier)と、繰り返しを持った属性の繰り返しを識別するサブレコード情報から成る。

集合の内部構造をBackus Normal Form(BNF)記法で以下に示す。

```

<集合> ::= '{' {<集合要素>','<集合要素>}' '}'
<集合要素> ::= '{' '<RID>' {<属性情報>','<属性情報>}' '}'
<RID> ::= レコードの識別番号
<属性情報> ::= '{' '<属性名>' {<属性値>','<属性値>}' '}'
<属性名> ::= <attr_id> ':' '<occ>' {<属性値>','<属性値>}'
<attr_id> ::= 属性の識別番号
<occ> ::= 属性の繰り返しを識別する番号
  
```

下図2-1を例に内部構造がどのようにあらわされるかを示す。

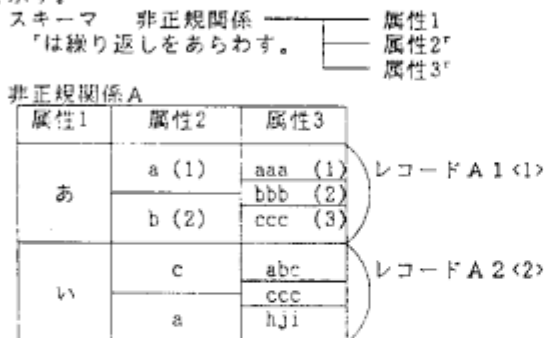


図2-1 非正規関係の例

これは、繰り返しを持たない属性(属性1)、2つの

繰り返しを持つ属性(属性2)と3つの繰り返しを持つ属性(属性3)からなる非正規関係である。

この非正規関係Aに対して次の検索を行った結果が図2-2の非正規関係Bである。

検索条件 = (属性2 = a or 属性3 = ccc)

非正規関係B

属性1	属性2	属性3	
あ	a	aaa	レコード B 1
		bbb	
あ	a	ccc	レコード B 2
	b		
い	c	ccc	レコード B 3
	a		
い	a	abc	レコード B 4
		hji	

図2-2 検索後の非正規関係

この非正規関係Bを集合の内部構造で表した結果が下図2-3である。例えば図2-3の1行目(1,({属性2:1}),({属性3:3})),(2,({属性2:2}),({属性3:2})))は図2-2のレコードB1,B2を表現している。

```

(1,({属性2:1}),({属性3:3})),
(2,({属性2:2}),({属性3:2})))
  
```

レコードの識別番号 属性2-1中の繰り返しの識別番号

図2-3 集合の内部構造

3. 集合要素とレコードとの対応付け

図2-3の集合と対応付けを行い図2-2の非正規関係Bとする処理を図2-1の非正規関係AのレコードA1を例に説明する。

- (1) 図2-3の集合から1番目の集合要素(1,({属性2:1}),({属性3:3}))を取り出す。
- (2) (1)で取り出した集合要素中のレコードの識別番号(RID)のレコードを読む。

この結果として図2-1の非正規関係AのレコードA1が読める。

- (3) (2)で読んだレコードA1と(1)で取り出した集合要素との対応付けをする。属性1は繰り返しを持たない属性なので図3-1中では-で表し、以降の処理では対象から除外する。

レコード A 1

属性1	属性2	属性3
あ	a	aaa
	b	ccc

属性1	属性2	属性3
-	1	1
	2	3

図3-1 集合要素との対応結果

Implementation of Set in Kappa DBMS

Tsutomu IGARASHI* Atsushi KANABGAMI** Moto KANAMURA***

*Mitsubishi Electric Computer System(Tokyo) Corporation **Mitsubishi Electric Corporation

***Institute For New Generation Computer Technology

- (4) (3)で集合要素中の繰り返しの識別番号と対応付けをした図3-1の非正規関係を正規形に分解し、図3-2の状態にする。

属性2	属性3
1	1
1	2
1	3
2	1
2	2
2	3

図3-2 正規形への分解結果

- (5) (4)で正規形に分解した図3-2から(1)で取り出した集合要素(1,({属性2:1}),({属性3:3}))を満たすレコードのみ取り出す。条件を満たすレコードを取り出したものが図3-3である。網かけした部分が条件を満たすレコードをあらわしている。

属性2	属性3
1	1
1	2
1	3
2	1
2	2
2	3

属性2	属性3
1	1
1	2
1	3
2	1
2	2
2	3

図3-3
条件を満たすレコードの取り出し結果

- (6) (5)で条件を満たすレコードを取り出した結果(図3-3)を非正規形にするため、(1)で取り出した集合要素中の属性を使い、繰り返し構造にまとめる順番を決める。

現在Kappaでは繰り返し構造にまとめる順番をスキーマ上の属性順にしている。この例の場合は属性2、属性3の順で繰り返し構造にする。

- (7) (6)で決定した順番に従い図3-3を非正規形にする。始めに属性2を繰り返し構造にする。図3-3の属性2以外の繰り返しの識別番号で重複しているものをまとめる。この例の場合は属性3の繰り返しの識別番号3が重複しているのでこれをまとめる。その結果図3-4の様に属性2が繰り返し構造となる。

属性2	属性3
1	1
1	2
1	3
2	

図3-4 属性2を繰り返し構造にした結果

次に属性3を繰り返し構造にする。図3-4の属性3以外の繰り返しの識別番号で重複しているものをまとめる。この例の場合は属性2の繰り返しの識別番号1が重複しているのでこれをまとめる。その結果図3-5の様に属性3が繰り返し構造となる。

属性2	属性3
1	1
1	2
1	3
2	

図3-5 属性3を繰り返し構造にした結果

図3-5の状態になるとこれ以上1つにまとめることは不可能なので2つの繰り返し情報に分けている。こうすることで正規形の意味を損なうことなく情報を返す事が出来る。

- (8) (1)~(7)の結果非正規関係の状態は図3-6となり、属性2、属性3共に繰り返し構造の属性となる。

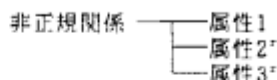


図3-6 属性2,3を繰り返し構造にした結果

ユーザに返すレコードは図3-5の繰り返し情報に従い(2)で読み出したレコードA1の値を対応付け図3-7の通り2つのレコードに分ける。

属性1	属性2	属性3
あ	a	aaa bbb
あ	a b	ccc

図3-7 レコードA1との対応付けを行った結果

これらの処理のうち(3)~(7)は拡張関係代数[2]の集合演算を集合を用いておこなったものであり、その結果を(8)でレコードに反映させている。このような処理が必要な理由は非正規関係とそれを正規関係にしたものが等しいと意味付けているためである。

4. まとめ

Kappaには繰り返しの種類としてset, list, bagがある。setは集合を意味し、繰り返しの要素の順番に意味は無く、要素間で値の重複が許されていない。listは繰り返しの順番に意味があり、要素間で値の重複が許されている。bagは繰り返しの順番に意味は無く、要素間で値の重複が許されている。しかし、現在はlist, bagに対する考慮が十分にされていない。今後はこれらの扱いについて更に検討を行う必要がある。

[参考文献]

- [1] Kazumasa Yokota, Moto Kawamura, Atushi Kanaegami "OVERVIEW OF THE KNOWLEDGE BASE MANAGEMENT SYSTEM (KAPPA)", PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON FIFTH GENERATION COMPUTER SYSTEM, 1988, Institute for New Generation Computer Technology(ICOT)
- [2] 昭和63年度版
電子計算機基礎技術開発成果報告書
-基本ソフトウェアシステム編-
財団法人新世代コンピュータ技術開発機構