データベースの最近の話題と医療への応用

Basis for Current Database Technology with Applications in Medicine

岡田美保子

Mihoko Okada

新潟大学教養部

Department of General Education, Niigata University

1. はじめに

大量の情報をコンピュータで操作するにはデータファイルを作成することが必要である。その場合、互いに関連する様々な応用業務が、目的ごとにそれぞれ独自のファイルを作成していったとすると、情報が重複して格納され無駄となるばかりでなく、ファイル相互の情報交換が簡単にできない、更新を重ねるうちにファイルの間で情報に矛盾が生じてしまう、などの問題が起きてくる。共通に利用される情報は1箇所にまとめて格納・管理し、これを様々な目的にスムーズに利用できるように便宜をはかるのがデータベース管理システムである。データベースそのものは、多目的に共有されることを前提として綿密に設計されたデータファイル群と考えられる。

構築されたデータベースに対しては、データの更新、情報検索、統計処理などの操作がなされる。これらの操作を汎用的な形でサポートすることは厳密にはかなり難しく、データベースは情報科学の中の一つの大きな研究テーマとなっている。データベースに関するテーマには物理的なレベルのもの（ディスク上でのファイル構築法など）から、現実の世界の対象（人や疾患、できごとなど）と、その相互関係をいかに表現するか、といった論理的なレベルまで様々なものが含まれる。後者は特にデータベースモデルの問題と言われている。

近年、人工知能の応用技術が進むにつれ、知識の処理と呼ばれる新しい情報処理の概念が普及してきた。データベース管理システムと知識処理システムとは当然密接に関連し、現在両者を統合した新しいシステム像が研究されている。こうした新しい情報処理の概念においては、情報の表現、情報の操作を論理的な段階でとらえることが重要である。本稿では、論理レベルのデータベースの

テーマとして、近年のモデルの中心であるリレーショナルモデルについて述べる。また論理型のプログラミング言語Prologを紹介しながら、推論を伴う情報検索や知識ベースの考えかた、そのリレーショナルモデルとの関連などについて述べたい。

2. リレーショナルモデル 表1 2項関係の例

2.1 関係 (relation)

リレーショナルモデル¹⁾の基本的な考え方は数学における関係 (relation) に基づいている。2つの集合AとBに対し、Aの要素と

科目	教室
生理学	A205
病理学	B510
解剖学	A205
生理学	C310
薬理学	C310

Bの要素の間の対応を約束する規則を2項関係 (binary relation) という。たとえば各教科目の使用教室に関する表1のような情報は、科目の集合A = {生理学, 病理学, 解剖学, 薬理学} と教室の集合B = {A205, B510, C310} との間の1つの2項関係として記述できる。相対応する2つの要素を(生理学, A205) のように対で表すと、2項関係はこのような要素対の集合により定義され、表1の情報を表す2項関係は

{(生理学, A205), (病理学, B510),

(解剖学, A205), (生理学, C310), (薬理学, C310)}

なる集合で表される。同様にしてn個の集合の上のn項関係 (あるいは単に関係という) が定義される。図1に示す4項目からなる情報は、科目、時限、教室、教授なる4つの集合の上の関係として表現することができる。この関係をcourseと名付けると、courseは

{(生理学, 水2, A205, 山本),

(病理学, 土1, B510, 田中), . . . }

なる4個の要素の組の集合により定義される。

図1 n項関係を記述する表

2.2 リレーショナルデータベース

リレーショナルモデルは現実世界の対象とその相互の関係を n 項関係により記述するもので、図式的に表形式で表される(図1)。表の各項目は属性(attribute)、各行はタプル(tuple)と呼ばれる。図1の表は4つの属性から成るが、表が表す関係courseの内容は属性の順序を入れ替えても変わりなく、courseは属性の集合 $R = \{\text{科目, 時限, 教室, 教授}\}$ の上で定義されるといえる。この属性の集合 R を関係courseのリレーションスキーマという。同様にして表の各行(タプル)は n 項関係を定義する集合の要素と考えられ、行の順序に特に意味はなく、また表の中に同一の行が2つ以上表れることはない。リレーショナルデータベースはこのような表の集合として記述される。

3. データベースの整合性

いま患者データの一部として、受診している診療科に関する情報が表2のように記録されていたとする。この場合、複数の診療科を受診している人については同一の患者情報(患者番号、氏名、住所)が診療科ごとに繰り返し格納されることになる。すると、たとえば番号100の人の住所が変更になった場合には、番号が100の行をすべて捜して書き換えなければならない。これに対し表3

表2 情報に重複を含んだ表

患者番号	氏名	住所	診療科	初診日
10	田中	三栄市	内科	75/5/10
50	鈴木	新潟市旭町	内科	80/12/5
100	佐藤	燕市	外科	79/11/28
10	田中	三栄市	皮膚科	83/10/10
200	山田	新潟市五十嵐	小児科	85/6/26
100	佐藤	燕市	内科	70/6/3

は同じ情報を2つの表に分けて記録したものであるが、この場合には住所変更による書き換えは1箇所だけで済み、表2よりも維持が簡単である。大規模データベースを長期にわたって維持してゆく場合、更新が常に正確に行えることは極めて重要であり、重複した情報はこれを保証する上で問題となる。データベースの操作上、より優れた表(関係)の構造を明らかにするため、リレーショナルモデルでは表の性質について数多くの形式的な研究がなされてきた。より良い性質を備えた形に表を導くことを関係の正規化という。

3.1 関数従属

表の性質を表す形式的な概念の1つに関数従属がある。これを考える前にまずキーの概念を定義する。患者ファイルに1患者につき1レコードが記録され、各レコードには患者番号が記されているとする。このファイルからデータを読む場合は、患者番号さえ指定すれば特定の1レコードを引出すことができる。しかし表2のようなファイルでは患者番号と診療科の値を指定しないと、どのレコードを指しているのかわからない。一般にファイルの中の特定の1レコードを指定するのに必要な項目の集りをキーという。リレーショナルモデルでは、一般に関係を r と表すとき、属性の集合 X については r のすべてのタプルが互いに相異なる値を持ち、かつ X の中のどの1つの属性を除いてもこの性質が成立しないとき、 X を関係 r のキーという。表2の関係では{患者番号, 診療科}がキーとなる。(キーは必ずしも1つだけではない。)あまり正確ではないが、およそキーは r のタプルが表す対象を集約する、あるいはキー以外の属性は意味の上でキーに従属する、などのように考えることができる。関係 r においては、キーの値が

表3 表2を分割した2つの表

患者番号	氏名	住所
10	田中	三栄市
50	鈴木	新潟市旭町
100	佐藤	燕市
200	山田	新潟市五十嵐

患者番号	診療科	初診日
10	内科	75/5/10
50	内科	80/12/5
100	外科	79/11/28
10	皮膚科	83/10/11
200	小児科	85/6/26
100	内科	70/6/3

決れば他のすべての属性の値はただ1つに定まる。このキーの概念を一般化したのが関数従属である。以下記号 X 、 Y は属性の集合を表す。 r の2つのタプルが属性 X について同じ値を持つならば、両者の値は Y についても必ず等しい(X の値が決れば Y の値はただ1つに決る)とき、 Y は関数的に X に従属するといい、この性質を $FD: X \rightarrow Y$ のように表す。たとえば表2では患者番号の等しいタプルは必ず氏名、住所も等しく $FD: \{\text{患者番号}\} \rightarrow \{\text{氏名}, \text{住所}\}$ が成り立つ。もし X がキーならば、スキーマ R のいかなる属性の値もただ一つに決ることから $FD: X \rightarrow R$ が成り立つ。

3. 2 関係の正規形 (normal form)

表2と表3の違いは関数従属を用いて説明される。表2のキーは{患者番号, 診療科}であり{患者番号, 診療科} $\rightarrow$ {氏名, 住所, 初診日} が成り立つ。しかし氏名と住所は患者番号にのみ依存し、{患者番号} $\rightarrow$ {氏名, 住所} が成り立っている。このとき{氏名, 住所}はキー{患者番号, 診療科}に部分従属するという。一般に $FD: X \rightarrow Y$ において、 X の真部分集合 X' に対し $X' \rightarrow Y$ が成り立つとき、 Y は X に部分従属するという。 $X' \rightarrow Y$ を満たすような X' は存在しないとき、 Y は X に完全従属するという。スキーマ R において、キーの中に含まれない属性はすべてキーに対して完全従属であるとき、 R は第2正規形(second normal form)にあるという。表2は第2正規形ではなく、これを分割した表3の2つの表はそれぞれ第2正規形にある。情報が重複している表の1つの性質が部分従属により説明される。

今度は表4の例をみる。キーは{ID番号}で、市町村、市町村コードはいずれもキーに含まれず、しかも{ID番号}に完全従属であるので表は第2正規形にある。しかし、たとえば三条市のコードが変わると、市町村が三条市であるすべてのタプルを書き換えなければならない。ここでは{ID番号} $\rightarrow$ {市町村}、{市町村} $\rightarrow$ {市町村コード}のように、{市町村コード}が{市町村}を介して{ID番号}に従属している。このような{市町村コード}の{ID番号}への従属のしかたを推移従属という。厳密な定義は省略するが、スキーマ R が第2正規形にあり、かつキーに含まれない属性はすべてキーに対し推移従属ではないとき、 R は第3正規形(third normal form)にあるという。表4は表5のように分割すれば、分割された表はそれぞれ第3正規形となる。推移従属も、情報に重複を生じるよう

表4 もう1つの重複を含んだ表

ID番号	氏名	市町村	市町村コード
10	田中	三条市	C350
50	鈴木	新潟市	C100
100	佐藤	燕市	C204
200	山田	新潟市	C100
210	小川	三条市	C350
220	中野	燕市	C204

表5 分割による重複の解消

ID番号	氏名	市町村	市町村	市町村コード
10	田中	三条市	三条市	C350
50	鈴木	新潟市	新潟市	C100
100	佐藤	燕市	燕市	C204
200	山田	新潟市		
210	小川	三条市		
220	中野	燕市		

な表を特徴づける概念で、やはり表分割により情報の重複が解消される。

データベースが正しく構築され、維持されることを保証するため、属性の性質について知り得ることから、記録される情報に対し何等かの約束(たとえば数値項目には数値以外が記されることはない)を設けることができる。リレーショナルモデルでは関係(relations)の整合性を維持するため、関係が満たすべき制約(constraints)を形式的な形で数々定義している。関数従属の概念はそうした制約の一つである。

4. データベースと知識ベース

近年人工知能の応用技術が注目を集めている。中でもエキスパートシステムは多くの応用分野で活発に研究開発され、一部ではすでに実用化もなされている²⁾。エキスパートシステムとは、ある専門領域において人の専門家と同様の知識を蓄え、専門家と同程度の作業(判断など)をすることができるコンピュータシステムをさす。エキスパートシステムがその専門的な作業を実行するために必要とする知識(情報)の集りを知識ベースと呼んでいる。以下簡単な例を用いて、この知識ベースの意味を考えてみたい。

4. 1 推論を伴うデータベースの検索

人工知能の応用領域では論理型のプログラミング言語Prolog³⁾が広く使われている。近年はデータベースの分野でも理論、応用の両面で同言語が導入されている^{4), 5)}。ここではProlog(インタ

表6 職員ファイルの例

氏名	職種	常／非
山本	医師	常
鈴木	医師	常
佐藤	医師	非
近藤	正看	常
加藤	正看	非
青木	準看	非
山田	理療	常
田村	薬剤	常

```

staff(山本, 医師, 常).
staff(鈴木, 医師, 常).
staff(佐藤, 医師, 非).
staff(近藤, 正看, 常).
staff(加藤, 正看, 非).
staff(青木, 準看, 非).
staff(山田, 理療, 常).
staff(田村, 薬剤, 常).

```

図2 表6の職員情報のPrologによる記述

ブリタ型を想定)を用いて、推論を伴うデータベース検索の考え方を紹介する。

例として表6のような職員データを考える。図2はこれをPrologで記述したものである。Prologを稼働してこのように入力すると、1つの簡単なプログラムがメモリ上に作成されたことになり、システムはプロンプト'?'-を表示してユーザの入力を待つ。ここで'staff'は適当につけた名前での文字列でもかまわない。このstaffの部分は述語と呼ばれ、かつこの中はその引数と呼ばれる。図2のプログラムはstaffなる述語を用いて1つの関係(データベース)を定義していると考えられ、ユーザはシステムに対してその内容に関する問い合せをすることができる。たとえば

```
?-staff(佐藤, 医師, 非).
```

と入力すると(下線はユーザの入力部)、この情報がデータベースに格納されているかを問うことになる。システムは検索を実行してこれを見つけ、yesと答える。図3に質問応答の例を示す。アルファベットの太文字で始る記号列は変数を表し、

```
?-staff(X, 医師, 常).
```

なる入力、変数Xにあてはまる値(すなわち常勤の医師は誰か)を尋ねることになる。システムはXにあてはまる値(答え)を1つ捜して返す。

そこで';'を入力すると、さらに他の答えを捜して

返してくる。もし他に答えがなければnoと返す。また最後の質問で用いている記号'_'は、その部分の値は何でもよい(無視してよい)ことを表す。

いま各職員の病歴データへのアクセス権について次のような規則があったとする。

「医師はすべてのデータにアクセスでき、その

他の常勤の職員はサマリにアクセスできる。」

この規則を付加したプログラムを図4に示す。記号':-'は「もし右に記されていることが成り立つならば、左のことは成り立つ」と解釈され、

```
access(X, all):-staff(X, 医師, _).
```

なる記述は「もしXが医師ならば、Xのアクセス権はallである」というルールを表すことになる。':-'の左の部分の頭部、右側の部分を本体という。ルールに対し、':-'を含まない記述を事実という。図4のプログラムを入力した後、たとえば

```
?-access(鈴木, What).
```

と入力すると、システムはそのような事実、またはこれにマッチする頭部を持ったルールがないか、データベースを先頭から順次検索する。するとこの質問はaccess(X, all):-staff(X, 医師, _)なるルールの頭部に(X=鈴木, What=allとおけば)一致する。そこで今度はstaff(鈴木, 医師, _)が真か否かを検索にゆく。これは事実として見出され、結局access(鈴木, What)はWhatにallを代入すると成り立ち、Whatの答えとしてallが返される。

この実行過程は、
a) Xが医師ならば
Xのアクセス権はallである。
b) 職員'鈴木'は医師である。
c) ゆえに職員'鈴木'のアクセス権はallである。
という推論を行ってアクセス権に関する問い合せに答えたと考えられる。

```

staff(山本, 医師, 常).
staff(鈴木, 医師, 常).
staff(佐藤, 医師, 非).
staff(近藤, 正看, 常).
staff(加藤, 正看, 非).
staff(青木, 準看, 非).
staff(山田, 理療, 常).
staff(田村, 薬剤, 常).
access(X, all):-
    staff(X, 医師, _).
access(X, summary):-
    staff(X, Job, 常),
    Job==医師.

```

図4 アクセス権に関するルールの記述。記号/=はnot equalを表す。

実際には一般法則から推論を行っていても、ユーザの立場からは、あたかも属性'アクセス権'の値が各職員ごとに個別具体的に記された表があるように見える。すなわち、一般法則からvirtual relationが構成されていると考えられる。個々のタプルごとに具体的な値として記されたデータベースをextensional databaseと呼ぶのに対し、一般法則(virtual relation)からなるデータベースをintentional databaseと呼んでいる。リレーショナルデータベースには本来Prologの推論に対応する検索機能はなく、上記のアクセス権情報は具体的な表の形で明示しなければならない。しかし近年は推論に相当する機能を持たせたりレシヨナルシステムの研究がなされている^{8), 7)}。

4. 2 構造を持った情報の記述

専門知識と呼ばれるものには通常（部分的にせよ）何等かの体系があり、専門家はその体系に基づいて自分の知識を構成しているものと考えられる。たとえば疾患の概念や医薬品の概念のように、体系をなす知識を計算機に格納するには、その体系の構造を表現しなければならない。知識表現は人工知能応用における中心テーマの1つであり⁸⁾、医学領域においても様々な表現方法が提案されている⁹⁾⁻¹²⁾。

例として図5のような医薬品の分類構造を考える。グラフに表された情報の記述単位をノードと呼ぶと、このような情報は個々のノードに関する情報の詳細と、各ノードのつながりを表す情報で記述される。このノード間の関係は、表7のような単純な表形式で表すこともできる。属性PARENTは各ノードの親ノードを表す。階層構造をなす情報では、その階層関係を表すことにより格納される情報の重複を防ぐことができる。すなわち上位のノードについて成り立つ性質、たとえば向精神薬一般について言えることは、向精神薬のノード1箇所にまとめて記載すればよい。その個々の下位ノードには、トランキライザならばトランキライザ固有の性質だけを記載すればよい。そしてたとえばマイナトランキライザの性質を求めるとき、そのあらゆる上位ノードの性質を抽出すればよい。

この場合、マイナトランキライザの直接の親ばかりでなく、親の親、そのまた親などをすべて求めるが必要になる。図6にこのような検索を行うためのプログラムを示す。parent(X,P)は「PはXの親である」ことを表し、predecessor(X,P)は「PはXの上位カテゴリである」ことを表す。predecessorを定義する最初のルールは「PがXの親ならば、PはXの上位ノードである」ことを

図5 医薬品の分類構造

表7 図5の階層関係の記述

NODE-NAME	PARENT
中枢神経抑制剤	none
催眠鎮静剤	中枢神経抑制剤
向精神薬	中枢神経抑制剤
フェノチアジン類	中枢神経抑制剤
トランキライザ	向精神薬
マイナトランキライザ	トランキライザ
マイナトランキライザ	催眠鎮静剤
⋮	⋮
⋮	⋮
⋮	⋮

```
parent(中枢神経抑制剤, none).
parent(催眠鎮静剤, 中枢神経抑制剤).
parent(向精神薬, 中枢神経抑制剤).
parent(フェノチアジン類, 中枢神経抑制剤).
parent(トランキライザ, 向精神薬).
parent(マイナトランキライザ, トランキライザ).
parent(マイナトランキライザ, 催眠鎮静剤).
predecessor(X, P):-
    parent(X, P).
predecessor(X, P):-
    parent(X, Y).
    predecessor(Y, P).
```

図6 親ノードと上位ノードの定義

```
?-predecessor(マイナトランキライザ, P).
P=トランキライザ;
P=催眠鎮静剤;
P=向精神薬;
P=中枢神経抑制剤;
P=none;
P=中枢神経抑制剤;
P=none;
no
```

図7 上位ノードの検索例

宣言する（これをrule1とする）。次のルールは「何か(Y)がXの親で、その何か(Y)の上位ノードがPならば、PはXの上位ノードである」としている（これをrule2とする）。この定義ではルールpredecessorを定義するのに、predecessor自身を用いている。このような定義の仕方を再帰的（recursive）定義という。このプログラムに対し ?-predecessor(マイナトランキライザ, P). と質問すると、図7に示すようにPの値が次々と返ってくる。この検索実行手順は次のようになる。まず質問とrule1の頭部がマッチして、その本体parent(マイナトランキライザ, P)が実行され、最初のP=トランキライザが返る。次の答えを要求するとparent(マイナトランキライザ, P)の他の答えを捜してP=催眠鎮静剤が返る。parent(マイナトランキライザ, P)にはもう他に答えがないので次の答えの要求に対しては rule2が適用され、その本体

parent(マイナトランクライザ', Y), predecessor(Y, P) が実行される。まず Y = トランクライザ' が求まり、続いて predecessor(トランクライザ', P) が実行される。ここであらためて rule1 が適用され、parent(トランクライザ', P) の答えとして P = 向精神薬 が返る。(以下省略。)

同様にして、個々のノードの性質がデータベースに記載されていたならば、マイナトランクライザ' のあらゆる上位ノードの性質を収集するプログラムを考えることができる。このような情報検索の仕組を Knowledge inheritance¹³⁾ (構造を通じて上位概念の性質を受け継ぐ) と呼ぶことがある。

4. 3 知識ベース

エキスパートシステムが用いる情報は、データベースに明示的に記された事実、推論により見出される事実、その推論を行うのに用いるルール、さらには答えを導くには、どのルールをどのような手順で利用したらよいか、といった手続きに関するものなどが含まれる。これらをすべて合せて知識ベースと呼んでいる。

5. おわりに

エキスパートシステムが専門的な作業を成し遂げるには知識の検索を迅速に行うことが必要で、そのためにはデータベースの管理技術が要求される。また従来のデータベース管理システムにとっては、エキスパートシステムの技術を応用することにより、高度な推論を伴う検索、質問応答を実現することが可能となる。近年はエキスパートシステム (ES) とデータベース管理システム (DBMS) を統合した、エキスパートデータベースシステム (EDS) と呼ばれる新しいコンピュータシステムの概念が生まれ、研究が始まっている⁵⁾。

医療データベースについては残念ながらまだ理想的なモデルは存在しないと見え、現在の段階では利用目的がかなり限定されている。医療データベースの難しさについては過去数々の議論がなされ、問題点としてたとえば a) 病歴データの標準化 (病歴情報の共有) の困難、b) 欠損データの処理、c) 時間軸にそった情報の表現方法、d) 画像、波形などの格納方法、e) プライバシーの保護、などがあげられている。現在、時間の概念を導入したりレーショナルモデルの研究や、画像処理プロセッサを組み込んだエキスパートデータベースシステムの研究などが進行中であり、本当の意味での多目的利用をサポートできるデータベースの実現は1990年代の課題であるといえる。

参考文献

- 1) D. Maier: The Theory of Relational Databases, Computer Science Press, Maryland, 1983.
- 2) Expert Systems in Engineering (Special issue), Computer, July, 1986.
- 3) I. Bratko: Prolog Programming for Artificial Intelligence, Addison-Wesley, Reading, Massachusetts, 1986.
- 4) I. Futo, F. Darvas and P. Szeredi: The application of PROLOG to the development of QA and DBM systems, in "Logic and Data Bases," H. Gallaire and J. Minker (Eds.), Plenum Press, New York, pp.347-376, 1978.
- 5) L. Kerschberg (Ed.): Expert Database Systems, Proc. the First International Workshop, Benjamin/Cummings, Menlo Park, California, 1986.
- 6) R. Reiter: Deductive question-answering on relational data bases, in "Logic and Data Bases," H. Gallaire and J. Minker (Eds.), Plenum Press, New York, pp.149-178, 1978.
- 7) C. L. Chang: DEDUCE2: Further investigations of deduction in relational data bases, in "Logic and Data Bases," H. Gallaire and J. Minker (Eds.), Plenum Press, New York, pp.201-236, 1978.
- 8) J. Mylopoulos and H. J. Levesque: An overview of knowledge representation, in "On Conceptual Modelling," M. L. Brodie, J. Mylopoulos and J. W. Schmidt (Eds.), Springer-Verlag, pp.3-17, 1984.
- 9) E. H. Shortliffe: Computer-Based Medical Consultations: MYCIN, Elsevier, New York, N.Y., 1976.
- 10) R. Davis, B. Buchanan and E. Shortliffe: Production rules as a representation for a knowledge-based consultation program, Artificial intell., Vol.8, pp.15-45, 1977.
- 11) F. Gomez and B. Chandrasekaran: Knowledge organization and distribution for medical diagnosis, IEEE Trans. Syst., Man, Cybern., Vol. SMC-11, pp.34-42, 1981.
- 12) M. Okada and M. Okada: Information management of drug-drug interactions based on a formal representation, IEEE Trans. Syst., Man, Cybern., Vol. SMC-16, pp.358-364, 1986.
- 13) R. J. Brachman: What IS-A is and isn't: An analysis of taxonomic links in semantic networks, Computer, Vol.16, pp.30-36, 1983.