データベースシステム　119－25
（１９９９．７．２２）

図形タイプのデータベースへの導入とその応用
－序論－

牧之内顕文　黒木進
e-mail: { akifumi, kuroki }@is.kyushu-u.ac.jp
九州大学大学院システム情報科学研究科
812-8581 福岡市東区箱崎 6-10-1

要旨

一般的な図形タイプをデータベースに導入する．図形タイプを位相空間表現である凸胞複体（とその拡張）で表す．凸胞複体上の演算子と述語とを定義し，それらを使って一般のデータベースで使われている属性を 2 次元図形属性で表現する方法とそれらを使った問合せ例について述べる．一般に，本稿の方法は「広がりのあるデータ」の表現に有効であると考える．

Enhancing Databases by Introducing Figure Datatypes
– A Preliminary Consideration –

Akifumi MAKINOUCHI　Susumu KUROKI
e-mail: { akifumi, kuroki }@is.kyushu-u.ac.jp
Graduate School of Information Science and Electrical Engineering
Kyushu University
6-10-1 Hakozaki, Higashi, Fukuoka 812-8581, Japan

Abstract

Figures are often used in bussiness as well as academic worlds to represent data. Figures are also important to represent spatial objects such as geographical entities. Two dimensional spatial objects like point, line, and polygon are actually target of datatype standardization for databases. This paper proposes figure datatypes for more general usage and shows how they are useful to represent data having extention.

1　はじめに

空間データベースの研究には長い歴史がある．このデータベースでは1次元，2次元，さらに3次元空間中の図形を扱う．これに時間軸を導入した時空間データベースでも同様に図形を扱う．

図形と言えば，私の知る限り，データベース界では永い間このような空間データベース関連の分野のみで議論されてきた．しかし，データベースを離れてみると，図形はオブジェクトの表現としていろいろなところで利用されている．例えば，「図表で」示すという時の図は，棒グラフなどのグラフである．又，何かの抽象的性質を表すのにも図形が用いられる．例えば，図1に示すのは人の性質をグラフ化したものである．このような図形表現は多分に主観的であるが，何かの情報を運ぶという意味で価値が（ことも）あろう．

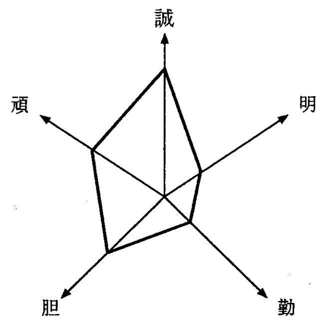

図 1. 人の性格付け 5 軸で表現

一方，空間データを対象としない現在の通常のデータベースでもよく使われる属性を考えると，図形として表現されるのが妥当だと思えるものがある．それは，一言で言えば「広がりのあるデータ」である．例えば，人の年齢をとりあげる．データベースの教科書によくある例として，学生とか従業員または社員データベースには「年齢」の属性が必ずといっていいほど存在する．関係表 教授 (氏名, 年齢) がある時，そのインスタンスとして，例えば ⟨牧之内顕文, 55⟩ としたとしても本人からクレームが来ることはないが，⟨斉藤良子, 58⟩ としたら問題が発生しないとは言えないだろう．⟨斉藤良子, $NULL$⟩ が政治的に正しい表記法である．ただし，ここでの NULL の用法は，既知だが，公表しないというぐらいの意味で使われる．⟨斉藤良子, 40 ～ 50⟩ としたら，本人からクレームが付くだろうか．なお，ここでの“斉藤良子”は全くの架空人物であることを断っておく．しかし，関係表 死亡不明者 (推定性別, 発見場所, 推定年齢) とした場合，そのインスタンスとして ⟨男, 博多駅, 15 ～ 60⟩ とされることがほとんどである．この場合，推定年齢は必ず「幅」を持つ．この幅は，1次元における広がりである．この場合，発見場所も2次元の広がりを持つが，この場合にはかなり局所的である（ので，2次元地図上の「点」で表現されるかもしれない）．

時間的推移が考慮される場合，値はかならず「広がり」を持つようになる．時間データモデル（Temporal Data Model）では時間は連続的に推移する．時間は，valid time と transaction time の2種類の時間で表記される．図2は [1] にあるビデオレンタルデータベースから取った例である．

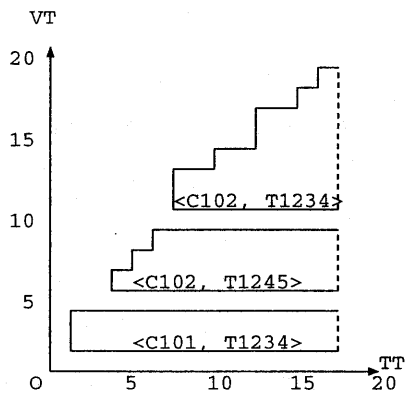

図 2. ビデオのレンタル DB（文献 [1] の表記とは若干異なる）

この例では，レンタル期間というデータを表現しており，時間を内在している属性であるといえる．しかし，「年齢」はどうだろう．これも，時間と共に変化するため，時間変化を内在している．体重はどうか．氏名はどうか．いずれも，強弱はあるが，時間（とともに変化する環境の変化）に関係し，その値が時間とともに変化しうる属性である．従って，これらは，時

間軸を横軸にして，縦軸にその属性値の値域（Domain）に当てると2次元上の図形として表現される．

本稿は，このように「広がり」のある値を図形で表現するデータタイプとその上の演算子を定義して，それらを使って，今まで考えられなかったデータベースをどう実現するかについて論じる．なお，本稿は「序論」であり，形式的な議論より，例示による説明となっていることを断っておく．

2 位相空間としての図形

「はじめに」で述べたような一般的な図形を定義するのにわれわれは位相幾何学での位相空間の一つの表現である凸胞複体 [2] を用いる．さらに、これを拡張した拡張凸胞複体 [3] も定義する．

2.1 凸胞複体

位相幾何学の概念である凸胞複体を用いて 4次元以下の空間の図形を表現する．凸胞複体により，0自由度の図形（点データ）や1自由度の図形（折れ線），2自由度の図形（多角形），3自由度の図形（多面体），4自由度の図形（超多面体）を統一的に表現できる．ここで，N次元空間 R^N（但し，R は実数，N は自然数）の超平面 H を，

$H = \{(x_1, \ldots, x_n) | \sum_{i=1}^{N} a_i x_i = 0$，ただし，係数$a_i$のいずれかは0でない$\}$

と書くと，H は点集合

$HS^+ = \{(x_1, \ldots, x_n) | \sum_{i=1}^{N} a_i x_i > 0$，ただし，係数$a_i$のいずれかは0でない$\}$

および

$HS^- = \{(x_1, \ldots, x_n) | \sum_{i=1}^{N} a_i x_i < 0$，ただし，係数$a_i$のいずれかは0でない$\}$

の境界となっている．一般に，ある図形 A を含む測度が最小の閉集合 A^a は閉包と呼ばれる．図形 A の閉包 A^a は，

- 図形 A の内点
- 図形の A の境界点

すべてからなる点集合である．これらの点集合 HS^+，HS^- の閉包 $\bar{HS}^+ = HS^+ \cup H$ と $\bar{HS}^- = HS^- \cup H$ を，それぞれ H を境とする半空間という．

このとき，R^N の有限個の半空間の交わりとして表される有界閉集合を凸胞という．凸胞を含む R^N のアフィン部分空間の次元の最小値 d を凸胞の次元といい，d- 凸胞と書く．

たとえば，点は 0- 凸胞，線分は 1- 凸胞，三角形や長方形などの凸多角形は 2- 凸胞，三角錐や立方体などの凸多面体は 3- 凸胞となる．凸胞を構成する半空間の境の集合と凸胞自身の積集合はまた凸胞である．このようにある凸胞 τ の境界の一部を構成する凸胞 f を一般的に辺という．凸胞 τ の辺である凸胞 f の辺もまた凸胞 τ の辺であると定義する．また，凸胞 τ 自身も辺であると定義する．これにより，たとえば凸胞 τ として直方体を考えると，凸胞 τ の辺は1個の直方体（3- 凸胞），6枚の長方形（2- 凸胞），12本の稜線（1- 凸胞），8個の頂点（0- 凸胞）である．

このとき，凸胞複体 C とは以下の条件を満たす凸胞の集合である．

1. 凸胞 τ が C の要素であれば，凸胞 τ の辺もまた C の要素である．
2. C の要素である任意の凸胞 τ_1 と凸胞 τ_2 の共通部分は空集合であるか，凸胞 τ_1 と凸胞 τ_2 のそれぞれの辺となっている．

図3に，1例を示す．

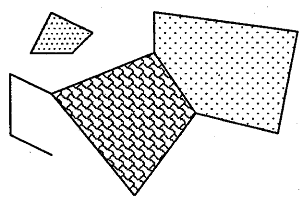

図 3. 凸胞複体によって表現される多面体

一方，凸胞複体は辺のつながりと同時に凸胞の頂点の座標データも持つため，図形の計量的な性質も表現している．頂点の座標データを用いると，図形の大きさや面の角度などを計算できる．このように位相的性質と計量的性質の両方によって凸胞複体が表現している図形を多面体という．

2.2 拡張凸胞複体

2.1で定義した凸胞に境界の開閉と向きを導入したのが拡張凸胞複体である．紙数の都合上，定義は省略する（[3] を参照）．この拡張凸胞複体によって，開線分，向きのある線分，面の裏表が表現できる．図 2の点線は「開」である境界を示している．

3 凸胞複体の基本演算子

前節で，凸胞複体の概念を定義した．また，図形の向きや開閉を導入し，拡張凸胞も定義できることを述べた．ここでは，凸胞複体に関わる基本演算子を導入する．拡張凸胞複体に関わるそれらはここでは触れない（紙数の関係上．詳しくは [3] を参照）．

3.1 集合演算子

凸胞複体は空間中の点の集合である．従って，2つの凸胞複体の点集合としての集合演算子を定義できる．結果は点集合であるが，それはやはり凸胞複体であることが保証される（ただし，その点集合が構成する一つの図形全体を凸胞に分割する方法は一意ではない）．以下，R^N 空間中の2つの凸胞複体をそれぞれ A, B とする．

$Intersection(A,B)$: A と B との共通部分集合を作る．結果は図形であり，ひとつの凸胞複体となる．空集合も凸胞複体である（便宜上．以下同じ）と定義する．

$Union(A,B)$: A と B との和集合を作る．結果は一つの図形であり，凸胞複体を構成する．

$Difference(A,B)$: A と B との差集合（A には含まれるが，B には含まれない点の集合）を作る．結果は一つの図形であり，凸胞複体を構成する．

$Section(A,B)$: A を B で決まる超平面（Hyperplane）で切断した時の断面を与える．B は超平面を唯一定義する 2 自由度の凸胞で表現される．一般的には 3 角形を構成する単体を指定すれば十分である．

$Projection(A,B)$: 点集合 B を含む次元最小のアフィン空間を C とする．このとき，A を C に射影して得られる点集合を与える．

いま R^3 の点集合 A と B を考える．B の表す点集合が線分であれば，C はその線分を含む直線である．また，B の表す点集合が一つの凸多角形であれば，C はその凸多角形を含む超平面である．

3.2 位相演算子

この演算子は，2つの凸胞複体を与えて，真偽値を与える演算子である．以下の演算子が考えられる．

$intersect(A,B)$: $x \in A$ でかつ $x \in B$ である点 x が存在するとき，真値を，それ以外は偽値を返す．上記集合演算子を用いると，もし $Intersection(A,B)$ が空集合でなければ真値を，それ以外は偽値を返すと定義できる．

$contain(A,B)$: $x \in B$ であるすべての点 x が $x \in A$ である時真値をその以外は偽値を返す（A が B を含む）．上記集合演算子を用いてこの演算子を定義すれば，$Difference(B,A) = \emptyset$ が成り立つとき真値を，それ以外は偽値を返す演算子である．

$meet(A,B)$: A と B との共通部分集合 $Intersection(A,B)$ がそれぞれの境界に含まれるならば真値をそれ以外は偽値を返す．「境界」という概念がこの定義には含まれるので，上記の（単純な）集合演算子では定義できず，この演算子の評価には，A と B の凸胞複体としての位相構造が必要となる．

$disjoint(A,B)$: A と B とが交差しないときは真値を，それ以外は偽値を返す．これは，言うまでもなく，$intersect(A,B)$ の否定である．

$equal(A,B)$: A と B とが互いに他の真部分集合であるとき，真値をそれ以外は偽値を返す．このことは，$contain(A,B)$ かつ，$contain(B,A)$ である．

4 凸胞複体データタイプ

前節で，凸胞複体の数学的な定義を述べた．この凸胞複体をデータベースのデータタイプとして利用するには，このままでは不十分である．例えば，2つの別個に定義された凸胞複体が存在しても，それらが「比較可能」であるとは限らない．これは，関係データモデルの任意の2つの関係表の積集合や和集合をとることができないのと同様である．ここでは，関係データモデルの「和互換（union compatible）」にならって，

二つの任意の凸胞複体の「座標互換（coordinate compatible）」を定義する．

2つの凸胞複体 A と B とは，次の条件を満たす時，座標互換（coordinate compatible）であるという．

(1) A と B とが定義されている座標空間の座標系が同じである．

(2) A と B とを同じ（対応する）座標軸に射影した時に得られる座標値は同じ値域（Domain）内にある．この値域という概念は，関係データモデルの Domain （属性の値域）と同じものである．

(1) について，たとえ同じ次元の座標系であっても，その方向が異なると互換にはならないことに注意する必要がある．例えば，2つの向きが異なる1次元座標の場合，論理的には同じ意味を表現するが，方向は逆向きである．従って，「座標互換」ではない．

(2) については，ある軸上でとりうる値が A と B とで同じ意味を持つことを保証するためである．例えば，X 軸が，A の場合には年齢を，B の場合は給料を表すということを排除する．

ここで注意すべきは，空間の座標軸は，数学的には実数軸であることである．しかし，データベースの世界では，属性の値は「実数」で表現されるものでなくてもいい．つまり，実数軸上で「離散」値をとりうる（ただし，この図形の世界では，属性値はすべて何らかの数値に変換されていると仮定する）．このような離散値の世界も，凸胞複体で表現できる．さらに，2つの離散値同士の間に線を引き，あたかも連続値であるかのように仮定して値を内挿することは可能である．ただし，その解釈は応用による．

2つの複体 A と B とが座標互換でないとしても，どちらかの複体をそれが定義されている座標空間の部分座標空間に写像されたとき，もう一方と座標互換になりうる．例えば，A が3次元座標空間 (X, Y, Z) で定義されており，B が2次元座標空間 (X, Y) で定義されているとする．A を部分座標空間 (X, Y) に射影したとき得られる複体は，もし両空間座標が上記 (1) と (2) を満足するなら，B と座標互換である．

5 応用例

凸胞複体データタイプの応用として空間データベースは第一の例である．本稿ではこのことに触れない．また，時空間データベースへの応用についても他 [2, 3] で論じたのでそれも省略する．ここでは，それ以外の応用例について述べる．

(1) 関係表 教授 (氏名, 年齢) で年齢を D(年齢) （但し，D は Domain の省略形）を軸とする一次元座標空間上の凸胞複体タイプとする．この空間上でとり得る凸胞は点か線分である．今，点で示せば，それはきりのいい年齢（例えば，45 才）であり，線分で示せば，幅のある年齢 （48 ～ 56 才）となる．（なお，線分で示される時は，年齢は「連続」していると考えていることになる．もし，この考え方が気に食わないとなれば，すなわち，年齢は離散的であると考えるならば，線分は点線に変わる．この場合の点線は，点の集合であって，異なる表示法をとった連続線分と考えるべきではない．）

この教授関係表に対して，「50 才くらいの教授」を検索するには，

```
SELECT 氏名
FROM 教授
WHERE intersect(年齢, 50 才くらい)
```

と書く．但し，「50 才くらい」は，年齢と「座標互換」のある一次元座標軸凸胞複体である線分で表現される．この問合せは，「50 才くらい」の線分と交わる年齢を持っている教授を選ぶことになる．

年齢を横軸に，氏名を縦軸にとれば，2次元座標空間 $(D(\text{氏名}), D(\text{年齢}))$ 上の凸胞複体でインスタンスを表現できる（図 4を参照）．

氏名年齢空間上の凸胞複体タイプを持つ属性「氏名年齢」を導入すると，教授関係表は教授 (氏名年齢) となり，氏名が B ～ C で年齢が 45 ～ 55 である教授を求める問合せは，

```
SELECT 氏名年齢
FROM 教授
WHERE contain(氏名年齢, B ～ C で 45 ～ 55)
```

となる．「B ～ C で 45 ～ 55」は図 4の長方形である．

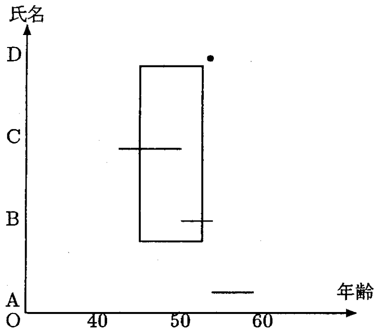

図 4. (D(教授), D(年齢)) 空間上の凸胞複体の例. A, B, C, D 各氏の年齢を表す図形ひとつひとつが凸胞（複体）. □は「B–C で 45 ～ 55」を示す図形.

(2) ここで, 教授 (氏名, 年齢, 友人) と「友人」属性で教授表を拡張する. 一人の友人は, 一次元名前空間上の点で表現し, 複数の友人は点集合で表現される凸胞複体である. 共通の友人を持つ教授を求めるには,

```
SELECT X. 氏名
FROM 教授 X, 教授 Y
WHERE intersect(X. 友人, Y. 友人)
```

のようになる. 同一の友人集合を持つ教授は

```
SELECT X. 氏名
FROM 教授 X, 教授 Y
WHERE equal(X. 友人, Y. 友人)
```

で求まる. “*equal*”を “=”で表現できるようにすれば,

```
SELECT X. 氏名
FROM 教授 X, 教授 Y
WHERE X. 友人 = Y. 友人
```

となり, 結合演算の集合への拡張となる.

(3) 上記の「友人」を横軸が名前空間, 縦軸が年齢とする 2 次元空間とすると

```
SELECT 氏名
FROM 教授
WHERE contain(友人, <大前健三, 50 才くらい>)
```

は, 名前が大前健三で 50 才くらいの友人を共有する教授の氏名を求めることになる. ただし, ここで注意すべきは, この 2 次元友人空間上の値は凸胞複体でなければならないので, この ⟨氏名, 年齢⟩ 凸胞（ここでは, 線分）が重なってはいけないことである. 即ち教授の友人には同姓同名がいなければ, この条件は満足される.

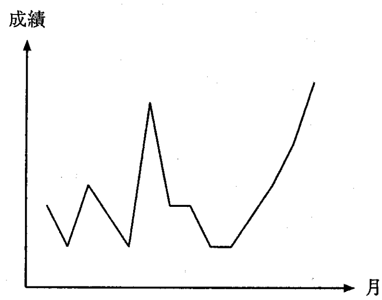

図 5. 月次営業成績を示す折れ線

(4) 1998 営業 (営業員番号, 1998 年度営業成績) を考える. ここでは, 1998 年度営業成績は月次の営業成績であり, 横軸が月, 縦軸が売り上げを示す 2 次元空間上の折れ線グラフであるとする（下図 5を参照）.

それは凸胞複体である. このデータベースに対して,「営業成績が前年度の営業成績を上回る営業員を求める」といった質問には次のような問合せを発行することになる. この場合, n 年度営業成績は座標互換であるのはいうまでもない.

```
SELECT 営業員番号
FROM 1998 営業, 1997 営業
WHERE 1998 営業. 営業員番号
       =1997 営業. 営業員番号
AND over(1998 年度営業成績, 1997 年度営業成績)
```

ここでの述語「over」は先に定義した位相演算子より定義することになる. どう定義するかは, たぶん会社の方針による.

6 おわりに

本稿では一般の図形をデータベース化するに必要なデータタイプについて論じた. 位相空間である凸胞複体を用いてこのデータタイプを表現する. データタイ

プの演算子として積，和，差などの集合演算子や，「交差する」，「含まれる」などの位相演算子（述語）を定義した．

また，凸胞に境界の開閉や向きを導入することについても触れた．それら図形タイプを使ったデータベース例と問合せ例をORDB表現を使って示した．もちろん，これらはODB表現と互換である．

今後はより厳密な定義を導入すること，さらにはより実用的な例とそれに伴って必要となる実際的な述語およびその実現法について検討する予定である．

参考文献

[1] Jensen, J., C. and Snodgrass, R., T.. Temporal Data Management, IEEE Knowledge and Data Engineering, Vol. 11, No. 1, pp.36–44 (1999).

[2] 黒木進，牧之内顕文．位相空間データモデルUniverseでの空間，時間，時空間データ表現，情報処理学会論文誌, Vol. 40, No. 5, pp. 2404–2416 (1999).

[3] 黒木進. 位相空間データモデルとその空間，時空間データベースへの応用に関する研究, 博士論文 (1999).