Vol. 30 No. 8　　情　報　処　理　　Aug. 1989

解　説

ファジィ集合論とソフトウェア†

馬　野　元　秀††

1. はじめに

ファジィ集合論[1]が提案されてから約25年がたち，実システムへの応用が進むにつれて，情報処理分野とも係わり合いができてきた．

現在，ファジィ情報処理という確立した分野があるわけではないが，ファジィ情報処理は大きく二つに分けることができそうである．一つはファジィ集合論に対する情報処理的アプローチであり，もう一つは情報処理の分野に対するファジィ集合論的アプローチである（情報処理というと，ハードウェアとソフトウェアに分けられるが，本稿ではソフトウェアとの係わりのみを考える）．

前者はファジィ集合論で定式化された種々の概念を計算機上に実現するためのもので，計算機によるファジィ集合の表現や処理の方法などがこれに当たる．後者は，われわれの行っている情報処理はきちんとしたものではなく，アルゴリズムもデータもあいまいであるのだから，これらをファジィ集合を使って定式化しようというものであり，ファジィ推論，ファジィ・プログラム，ファジィ・データベースなどがこれに当たる．

本稿では，これらについて，できるだけ多くの例を用いて説明する．以下，**2.**では，ファジィ集合とその演算についてまとめ，**3.**では，ファジィ集合の表現と処理，ファジィ推論，ファジィ・プログラム，ファジィ・データベースについて概説する．

2. ファジィ集合とその演算

2.1 ファジィ集合

ファジィ集合は，要素が集合に属するか属さないかを明確に定義できない場合に用いられる[1]．そのために，ある要素が集合に属する度合い（グレード）を考える．グレードは，完全に属するを1とし，完全に属さないを0として，その中間の値を考える．

[例1] 大きさの範囲として，0から10までの11段階を考える．このとき，「大きい」，「小さい」，「中くらい」は，たとえば，

大きい＝{0.1/6, 0.2/7, 0.8/8, 1/9, 1/10}，

小さい＝{1/0, 1/1, 0.8/2, 0.2/3, 0.1/4}，

中くらい＝{0.1/3, 0.8/4, 1/5, 0.8/6, 0.1/7}

と定義することができる．{ } の中では，グレード/要素の形式で表記している．■

もちろん，要素が連続でもよく，そのときには，グレードは式やグラフで与えられる．

[例2] 大きさの範囲を0から10までの連続区間とする．このとき，「大きい」，「小さい」，「中くらい」は，たとえば，**図-1**のように表すことができる．■

一般的には，ファジィ集合 A は考えている要素全体（全体集合）から単位区間 [0, 1] への関数 μ_A により定義され，この関数はメンバシップ関数と呼ばれる[1]．

さらに，要素が組になったファジィ関係[1]，グレードが組になったL-ファジィ集合[2]，要素がファジィ集合になったレベル2ファジィ集合[3]，グレードが区間 [0, 1] 上のファジィ集合になったタイプ2ファジィ集合[4]，これらを組み合わせた一般ファジィ集合なども提案されている．

2.2 ファジィ集合の演算

ファジィ集合の演算は，大きく分けて，以下の三つに分類できる．

（1） 通常の集合演算の拡張

和集合，共通集合，補集合などを求める演算で，ファジィ集合の場合には，

共通集合：$A \cap B = \{\mu_A(u) \wedge \mu_B(u)/u : u \in U\}$，　(1)

和集合　：$A \cup B = \{\mu_A(u) \vee \mu_B(u)/u : u \in U\}$，　(2)

補集合　：$\sim A = \{1 - \mu_A(u)/u : u \in U\}$　(3)

と定義される[1]．ここで，U はファジィ集合 A と B

† Fuzzy Set Theory and Software by Motohide UMANO (Computation Center, Osaka University).

†† 大阪大学大型計算機センター

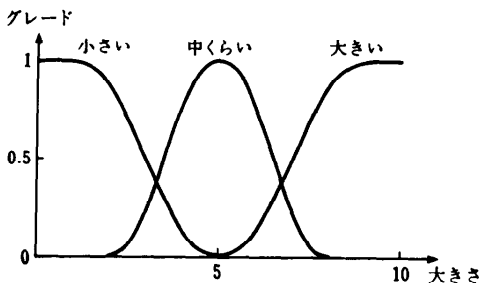

図-1 ファジィ集合「大きい」,「小さい」,「中くらい」

の全体集合であり，∧は min の演算を，∨は max の演算を表している．

[例3] 例1の「大きい」,「小さい」,「中くらい」を考える．このとき，

中くらいかまたは大きい＝中くらい∪大きい
＝{0.1/3, 0.8/4, 1/5, 0.8/6,
0.2/7, 0.8/8, 1/9, 1/10}
大きくない＝～大きい
＝{1/0, 1/1, 1/2, 1/3, 1/4,
1/5, 0.9/6, 0.8/7, 0.2/8}
小さくない＝～小さい
＝{0.2/2, 0.8/3, 0.9/4, 1/5,
1/6, 1/7, 1/8, 1/9, 1/10}
大きくも小さくもない
＝大きくない∩小さくない
＝{0.2/2, 0.8/3, 0.9/4, 1/5,
0.9/6, 0.8/7, 0.2/8}

となる． ■

この型の演算子は，二つ（以上）のファジィ集合の同じ要素のグレードに対して，min や max などの演算を行うものである（～A は $U-A$ と考える）．

なお，一般的には，∨と∧は，それぞれ，t-ノルム (triangular norm) と t-コノルム（または s-ノルム）という関数群に置き換えることができる(文献5)を参照)．さらに，これらの中間的な演算子として，平均演算子や補償演算子も提案されている(文献5)を参照)．

（2） 要素に対する演算の拡張

普通の要素に対して定義されている関数や演算をファジィ集合に適用できるようにするために，拡張原理が使われる[4]．

[拡張原理1] 全体集合 U における要素 u に対して定義されている1変数関数 f は U におけるファジィ集合 A の上に拡張して定義され，

$$f(A)=f(\{\mu_A(u)/u : u\in U\})$$
$$=\{\mu_A(u)/f(u) : u\in U\} \quad (4)$$

となる． ■

これは，すべての要素に関数を適用し，結果のグレードはもとの要素のグレードを採用するということを意味している．

[例4]「3くらい」を表すファジィ集合を

3くらい＝{0.3/2, 1/3, 0.5/4}

とする．このとき，「3くらい」の2乗は，

3くらい^2＝{0.3/2^2, 1/3^2, 0.5/4^2}
＝{0.3/4, 1/9, 0.5/16}

となる． ■

2変数関数に対しては，各ファジィ集合の要素のすべての組合せに関数を適用し，結果のグレードは各要素のグレードの min をとる．

[拡張原理2] 全体集合 U_1 における要素 u_1 と U_2 における u_2 に対して定義されている2変数関数 f は U_1 におけるファジィ集合 A と U_2 における B の上に拡張して定義され，

$$f(A, B)$$
$$=f(\{\mu_A(u_1)/u_1 : u_1\in U_1\}, \{\mu_B(u_2)/u_2 : u_2\in U_2\})$$
$$=\{\mu_A(u_1)\wedge\mu_B(u_2)/f(u_1, u_2) : u_1\in U_1, u_2\in U_2\} \quad (5)$$

となる．異なる u_1 と u_2 に対して，$f(u_1, u_2)$ が同じ値になるものが存在するときは，一つにまとめて，そのグレードは max で計算する． ■

[例5]「3くらい」と「4くらい」を表すファジィ集合を

3くらい＝{0.3/2, 1/3, 0.5/4}
4くらい＝{0.4/3, 1/4, 0.6/5}

とするとき，これらの足し算は

3くらい＋4くらい
＝{0.3/5, 0.3/6, 0.3/7, 0.4/6,
1/7, 0.6/8, 0.4/7, 0.5/8, 0.5/9}
＝{0.3/5, 0.4/6, 1/7, 0.6/8, 0.5/9}

となる． ■

A と B のいずれか一方だけがファジィ集合で，他方がファジィ集合でないときは，ファジィ集合でないほうを u とすると，それをその要素だけからなるファジィ集合 $\{1/u\}$ と考えて，拡張原理を適用する．

（3） グレードに対する演算

ファジィ集合のすべてのグレードに対して，ある関数を適用する演算であり，

$$f[A]=f[\{\mu_A(u)/u : u\in U\}]$$
$$=\{f(\mu_A(u))/u : u\in U\} \quad (6)$$

と定義される．補集合の演算～はこの型の演算とも考えられる．

[例6] 例1の「小さい」のグレードを2乗すると，

$$[\text{小さい}]^2=[\{1/0,\ 1/1,\ 0.8/2,\ 0.2/3,\ 0.1/4\}]^2$$
$$=[\{1/0,\ 1/1,\ 0.64/2,\ 0.04/3,\ 0.01/4\}]^2$$

となり，「とても小さい」を表すファジィ集合としてよく使われる[6]． ■

3. ファジィ集合とソフトウェア

3.1 ファジィ集合の表現と処理

ファジィ集合論を応用したシステムを作ろうと思うと，まずファジィ集合をコンピュータ上に表現・処理する部分のプログラムを作成しなければならない．ファジィ集合が使えるプログラミング言語があれば便利であるが，そのような言語はほとんどない．筆者が知るかぎり，ファジィ集合論的データ構造システム[7]と Lisp によるファジィ集合処理システム[8),9]だけである．

（1） ファジィ集合論的データ構造システム

ファジィ集合論的データ構造システム[7]は，ファジィ集合を表現するためのデータ構造を保持しており，約 50 個のファジィ集合演算子をもっている．通常のファジィ集合だけではなく，ファジィ関係，L-ファジィ集合，レベルmファジィ集合，タイプnファジィ集合，一般ファジィ集合を表現することができる．このシステムは，Fortran によりインプリメントされており，Fortran 中に埋め込んだ形でも使用できる．

しかし，このシステムを知識情報処理の分野に応用するには，機能が不十分であったので，このシステムを発展させて，Lisp によるファジィ集合処理システムが新たに作成された．

（2） Lisp によるファジィ集合処理システム

Lisp によるファジィ集合処理システム[8),9]では，ファジィ集合はリストとして表現されている．このシステムの特徴としては，（1）のシステムの特徴に加えて，

① ファジィ集合は｛ ｝を使って，組 (tuple) は〈 〉を使って自然な形で書ける

② 自分でファジィ集合用の関数を容易に定義できる

③ Lisp の関数をファジィ集合や組にも適用できる（拡張原理による計算）

④ Lisp のほとんどの機能をそのまま使用できるなどがある．

以下，このシステムの機能を実行例を用いて示そう．

[例7] 記号 F1 と F2 にファジィ集合を代入するには，Lisp の関数 setq を用いて，

```
-> (setq F1 {0.1/a, 0.2/b, 0.3/c})
{0.1/A, 0.2/B, 0.3/C}
-> (setq F2 {0.7/b, 0.2/c, 0.4/d})
{0.7/B, 0.2/C, 0.4/D}
```

とすればよい．ここで，-> はシステムからのプロンプトである．そして，これに続いて下線を引いた部分をユーザが入力すると，システムがユーザの入力を評価し，次の行にその値を表示する． ■

[例8] 例7の F1 と F2 の和集合と共通集合を求めるには，

```
-> (union F1 F2)
{0.1/A, 0.7/B, 0.3/C, 0.4/D}
-> (intersection F1 F2)
{0.2/B, 0.2/C}
```

とすればよい． ■

これら以外にも，約 60 個の組込み関数がある．

[例9] Lisp の関数は，前に & を付けることによりファジィ集合に対して適用できるようになり，計算は拡張原理で行われる．

例4と例5の計算は，まず，ファジィ数

```
-> (setq about-3 {0.3/2, 1/3, 0.5/4})
{0.3/2, 1/3, 0.5/4}
-> (setq about-4 {0.4/3, 1/4, 0.6/5})
{0.4/3, 1/4, 0.6/5}
```

と2乗する関数

```
-> (defun sqr (x) (* x x))
sqr
```

を定義し，

```
-> (& sqr about-3)
{0.3/4, 1/9, 0.5/16}
-> (&+ about-3 about-4)
{0.3/5, 0.4/6, 1/7, 0.6/8, 0.5/9}
```

とすればよい． ■

[例 10] 例6の計算は，例9で定義した関数 sqr を用いて，

```
-> (gapply 'sqr
          {1/0, 1/1, 0.8/2, 0.2/3, 0.1/4})
{1/0, 1/1, 0.64/2, 0.04/3, 0.01/4}
```

とすればよい． ■

現在，このシステムは Franz Lisp (Unix 4.X BSD)，UTI-Lisp (ACOS-6/MVX)，muLisp-86

(MS-DOS), Kyoto Common Lisp (Unix 4.X BSD) 上で，稼働している．

このようなファジィ集合を記述できるプログラミング言語は，非常に重要であると思われるが，他にほとんど存在していない．今後のファジィ集合の利用を促進するためにも，このようなプログラミング言語が数多く作成され，広く普及することが望まれる．

3.2 ファジィ推論

われわれが行う推論は，あいまいなルールとあいまいなデータ（事実）に基づいている．しかし，推論過程のどの部分をどのようにファジィ化するかによって，いくつかの方法がある．

（1） 確 信 度

確信度 (certainty factor) は，最初に MYCIN[10]でプロダクションシステムとともに使われて以来，知識工学の分野で広く用いられている．

確信度はデータやルールの結論部がどれくらい確かであるかを0と1の間の数値を使って表す（MYCINでは，−1と+1の間の数値で表している）．そして，確信度を用いた推論は，次のように行う[11]．

① ルールの個々の条件に一致するデータの確信度から，条件部全体としての確信度を求める．これには，たとえば，最小値を求める演算が用いられる．

② 条件部全体の確信度とルールの結論の確信度から，推論結果に対する確信度を求める．これには，たとえば，乗算が用いられる．

③ 二つ以上のルールが同じ推論結果を導き出すとき，それぞれの確信度から，その推論結果に対する全体的な確信度を求める．これには，たとえば，最大値を求める演算が用いられる．

ソフトウェアとして実現するには，通常の推論に確信度を計算する部分を追加するだけですむので，現在のエキスパートシステム構築用ツールでは，確信度が利用できるのが普通になっている．

現状では，「ファジィ推論」を使ったとか，「ファジィ・エキスパートシステム」であるといっているものでも，このレベルのものが多い．

（2） 近 似 推 論

あいまいな意味をもつ単語を含む推論，

関係：x と y はほぼ等しい
事実：x は大きい　　(7)
結果：y は？

ルール：もし x が大きければ y は小さい
事実：x はとても大きい　　(8)
結果：y は ？

を定式化しようとしても，確信度による方法ではまったく対処することができない．

L. A. Zadeh はファジィ集合の概念を用いてこのようなあいまいな推論を定式化した[4]．

式(7)の推論の一般的な形式，

関係：x と y は R である
事実：x は A である　　(9)
結果：y は B である

を考える．ここで，R は全体集合 $U\times V$ におけるファジィ関係で，A は U における，B は V におけるファジィ集合である．このとき，B は，

$$B=A\circ R$$
$$=\{\max_{u\in U}(\mu_A(u)\wedge\mu_R(u,v))/v : v\in V\} \quad (10)$$

により求めることができる．この式は「推論の合成規則 (compositional rule of inference)」と呼ばれる．

次に，ルールを用いた推論，

ルール：もし x が P ならば y は Q である
事実：x は A である　　(11)
結果：y は B である

について考える．ここで，P と A は全体集合 U における，Q と B は V におけるファジィ集合である．このとき，ルールは算術規則，

$$R=\{1\wedge(1-\mu_P(u)+\mu_Q(v))/\langle u,v\rangle : u\in U,\ v\in V\} \quad (12)$$

により $U\times V$ 上のファジィ関係 R に変換し，推論の合成規則(10)により，B を計算する．

[例 11] 式(8)のファジィ推論は，

$U=V=\{1,\ 2,\ 3,\ 4,\ 5\}$,
大きい$=\{0.6/4,\ 1/5\}$,
小さい$=\{1/1,\ 0.6/2\}$,
とても大きい$=\{0.3/4,\ 1/5\}$

とすると，算術規則(12)と推論の合成規則(10)により

とても大きい$\circ R$
$=\{1/1,\ 0.6/2,\ 0.3/3,\ 0.3/4,\ 0.3/5\}$

となる．この結果は，「まあ小さい」に近いと考えることができる． ■

その後の研究の結果，算術規則以外にも新しい変換規則がいくつか提案されており，推論の合成規則(10)の改良も提案されている（文献5)を参照)．

近似推論は，1段の推論過程としては非常によく研究されているが，ソフトウェアによる実現という観点

からみると，あまり進んでいない．それは，

① ルールの条件の数が多くなったときに，多くのメモリを必要とする上に，実行速度が遅い

② ルールがたくさん存在するときの実行方法があまり研究されていない

からである．

（3） ファジィ制御におけるファジィ推論

ファジィ制御は，オペレータの経験的な制御知識をファジィ集合を含むルールの形で記述し，ファジィ推論により制御動作を決定する方法である．

ファジィ制御のルールは，たとえば，

rl : if E＝ほぼゼロ & ΔE＝正で小さい
then ΔU＝負で小さい

のようなもので，「ほぼゼロ」，「正で小さい」，「負で小さい」などがファジィ集合である．このようなルールが数個あり，E と ΔE の値が分かったとき，①これらの値に対するメンバシップ関数の値として，条件の一致度を求め，②これから条件部全体の一致度を計算し，③それにより結論のファジィ集合を重み付け，④重みを付けた各ルールの結論を結合し，⑤一つの値を選び出す．もっともよく使われる推論法であり，文献[12]や本特集の他の記事でも紹介されているので，詳細については省略する．

この方法は，データにあいまいさはないが，ルールの条件と結論にあいまいさを入れることができる．しかし，各ルールの推論結果を結合するため，ルールの条件や結論はかなり制限された形になっている．

ソフトウェアとしての実現は，それほど難しくなく，かなり高速な処理も可能であり，汎用のファジィ制御用のツールも数社から発売されている．また，ハードウェアとして実現されている[13]のは，この推論方式である．

（4） その他のファジィ推論法

これら以外に，Prolog やプロダクション・システムをファジィ化したファジィ Prolog[14)~17)] やファジィ・プロダクション・システム[18),19)] も提案されている．

ファジィ Prolog には，いろいろなものがあるが，文献[14)~16)]は確信度的にあいまいさしか取り扱っていない．ただし，文献[16)]における Prolog では，確信度として区間値を取り扱うことができる．一方，文献[17)]のものは，単語の意味がもつあいまいさと確信度的なあいまいさの両方を表現でき，ファジィ的なパターン・マッチングにより動作する（「若い」と「25 歳くらい」でも，ある一致度でマッチする）．

Prolog の場合，理論的な考察も重要である．確信度的のみを扱う場合は，理論的な取り扱いもそれほど難しくはないが，ファジィ的なパターン・マッチングを可能にすると，理論的な取り扱いがかなり難しくなる．

ファジィ・プロダクション・システム[18),19)] はファジィ集合を含むルールやデータを書くことができるだけでなく，確信度的なあいまいさも記述できる．そして，データと条件はファジィ的にマッチし，結論に任意の動作（たとえば，Lisp のプログラム）が書くことができる．その代わり，ファジィ制御の場合のように，推論結果を結合することはできず，ファジィなデータともっともよく一致するルールを選び，その動作部を実行する．

3.3 ファジィ・アルゴリズムとファジィ・プログラム

われわれが使うアルゴリズムやプログラムは，ルールだけでは記述できない．そこで，手続き型のアルゴリズムやプログラムのファジィ化の研究も行われている[20)~22)]．

（1） ファジィ・アルゴリズム

ファジィ・アルゴリズムは，Zadeh によりファジィ集合を含む代入文，動作文，条件文の列であると定義された[20)]．それぞれの文の例として，次のようなものがある．

① 代入文：$x \approx 5$，x＝small，x is large など

② 動作文：multiply x by y，decrease x slightly，delete the first few occurrence of x，stop など

③ 条件文：
if x is small then y is large else y is not large，
if x is positive then decrease y slightly，
if x is small then go to 7 など

そして，ファジィ・アルゴリズムを次の５つに分類し，それぞれに対して例をあげた．

① ファジィ定義アルゴリズム：(例)楕円の定義

② ファジィ生成アルゴリズム：(例)文字Ｐの生成
(例)チョコレート・ファッジ（お菓子の１種）の作り方

③ ファジィ関係アルゴリズム：変数間の関係を与える．ファジィ制御はこれの発展形と考えられる．

④ ファジィ行動アルゴリズム：状態遷移表の拡張

⑤ ファジィ決定アルゴリズム：(例)交差点を渡る．(例)障害物を避けて物体を移動させる．

これらの例の解釈・実行には，拡張原理と推論の合

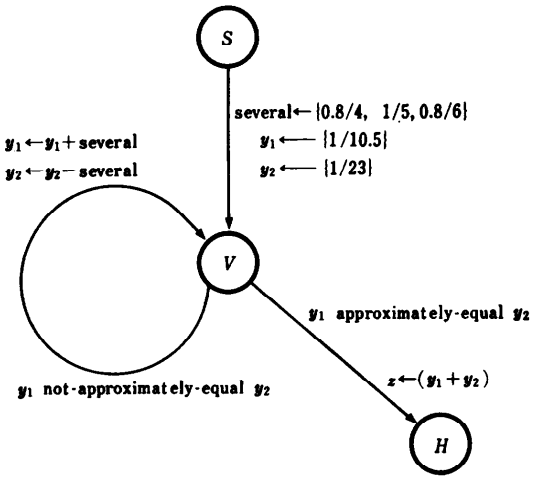

図-2 ファジィ・プログラムの例

成規則が使われるが，ad hoc 的な部分もかなり多い．

さらに，シミュレーションの例として，文字生成と車の運転手の行動（ファジィ命令と正確な地図により目標に到達する）がある[21]．

（2） ファジィ・プログラム

流れ図型のファジィ・プログラムが提案されている[22]．これは，(1)の特殊な場合に当たるが，通常のプログラムにより近い．

[例 12] 図-2 のようなファジィ・プログラムを考える[22]．ここで，approximately-equal はファジィ関係で，同じ要素に対してはグレードが1で，0.5 異なる要素に対しては 0.5 で，1 以上異なる場合には 0 とする．not-approximately-equal は approximately-equal の補集合である．

このプログラムの実行において，節点 V で分岐のどちらを選ぶかが問題となる．いまは両方のパスが並列実行されると考え，たとえば，条件 y_1 approximately-equal y_2 は，

$$y_1 \leftarrow y_1 \cap (\text{approximately-equal} \circ y_2)$$
$$y_2 \leftarrow y_1 \cap (y_1 \circ \text{approximately-equal}) \qquad (13)$$

と同じ効果をもつと考える．ここで，$\cap$ は共通集合の演算で，$\circ$ は式(10)の合成の演算である．この式は，それぞれの変数の値を条件が成立する部分だけに限定するということを意味している．また，ファジィ集合どうしの足し算と引き算は，式(5)の拡張原理2により計算する．実行結果をまとめると表-1 のようになる． ■

最近，ファジィ推論（ルール型のファジィ・アルゴリズムと考えられる）の研究はよく行われているが，

表-1 ファジィ・プログラムの実行

制御パス	y_1	y_2	z
S	—	—	—
SV	{1/10.5}	{1/23}	—
SVH	ϕ	ϕ	ϕ
SVV	{0.8/14.5, 1/15.5, 0.8/16.5}	{0.8/19, 1/18, 0.8/17}	—
$SVVH$	{0.5/16.5}	{0.5/17}	{0.5/33.5}
$SVVV$	{0.8/18.5, 0.8/19.5, 1/20.5, 0.8/21.5, 0.8/22.5}	{0.8/15, 0.8/14, 1/13, 0.8/12, 0.8/11}	—

手続き型のファジィ・アルゴリズムやファジィ・プログラムの研究はあまり行われていないようである．しかし，自然言語に基づくプログラミングへの可能性ももっており，非常に重要な分野の一つであるので，活発な研究が望まれる．また，重要なプログラミング・パラダイムの一つであるオブジェクト型のファジィ化も非常に興味深い．

3.4 ファジィ・データベース

現実の世界に存在するあいまいなデータを蓄積・検索・処理するために，ファジィ・データベースが提案されている．

（1） あいまいな質問

まず，通常の関係モデルに対して，質問の検索条件にファジィ集合を用いることが考えられる．

V. Tahani は，SEQUEL を基にして，あいまいな質問の処理方法を示した[23]．

[例 13] 図-3 のようなデータベースに対して「年齢が若いか最近採用された人で，給料の高い人の名前を求めよ」という質問を考える[23]．検索言語で書くと，

SELECT 名前
FROM 従業員
WITH (年齢＝young or 採用年＝recent)
and 給料＝high

となる．ここで，young, recent, high はそれぞれの属性におけるファジィ集合である．

従業員

名 前	年 齢	給 料	採用年
Anderson	30	20000	1974
Brown	30	15000	1974
Long	25	40000	1972
Nelson	55	20000	1950
Smith	25	25000	1975

図-3 従業員データベース

表-2 ファジィな質問の処理

組＼条件	年齢＝young	採用年＝recent	給料＝high	質問全体
⟨Anderson, 30, 20000, 1974⟩	0.5	0.6	0.5	0.5
⟨Brown, 30, 15000, 1974⟩	0.5	0.6	0	0
⟨Long, 25, 40000, 1972⟩	1	0	1	1
⟨Nelson, 55, 20000, 1950⟩	0	0	0.6	0
⟨Smith, 25, 25000, 1975⟩	1	0.8	0.8	0.8

従業員の各組に対して，条件との一致度を計算するが，ファジィ集合との一致度はメンバシップ関数の値とし，and と or は，それぞれ，min と max で計算する．各組に対する一致度を計算すると，表-2 のようになる．そして，質問に対する結果は，必要な定義域だけを取り出し，

{0.5/Anderson, 1/Long, 0.8/Smith}

となる． ■

（2） データ・モデルの拡張

あいまいなデータを表現するには，従来のデータ・モデルでは不十分なので，データ・モデルを拡張しなければならない．ほとんどのものは関係モデルを拡張したものになっているが，どのような種類のあいまいさを表現しようとするかによって，定式化のしかたはかなり異なっている．

① ファジィ関係モデル：もっとも簡単な拡張は，通常の関係にグレードを付加してファジィ関係にすることである．いくつかのシステムで使われている[24),25)]が，これらは，ファジィ・データベースとしての研究が中心ではない．

② 類似関係に基づく統合：質問の条件に合う組を検索してから，等しい組を一つにまとめるが，類似関係[26)]により，よく似た組をまとめるようにしたファジィ・データベースが提案されている[27)]．このために，集合を要素とできるように関係モデルを拡張している．

③ 可能性分布-関係モデル：データ自身のもつあいまいさを可能性分布[28)]で表現し，それを属性値としてもつことができる可能性分布-関係モデルが提案され，これに基づくデータ操作言語もインプリメントされている[29),30)]．

[例 14] 図-4 のような可能性分布を含む関係を考える[29)]．リチャードの子供の名前 {ジュディ，アンナ}p が可能性分布で，ジュディとアンナのどちらか一方を値としてとることを表している．レイモンドの年齢は

人 間

名 前	年 齢	子供名
トム	23	テッド
スーザン	35	ジョン
スーザン	35	マイク
リチャード	40	{ジュディ，アンナ}p
レイモンド	若い	分からない
ビクター	分からない	なし
スミス	{50, 51}p	NULL

図-4 可能性分布を含む関係

「若い」という可能性分布であり，これは，たとえば，

若い＝{0.3/15, 0.6/16, 0.8/17, 1/18, 1/19, 1/20, 1/21, 1/22, 1/23, 0.9/24, 0.8/25, 0.7/26, 0.5/27, 0.3/28, 0.1/29}p

と定義できる．そして，「分からない (unknown)」や「なし (undefined)」や「NULL (unknown であるか undefined であるかも分からない)」も可能性分布として解釈することができる．

これに対して，「年齢が 25 歳以上の人をみつけよ」という質問をすると，質問の条件を「明らかに満たす」ものと「明らかに満たさない」もの以外に，「満たす可能性がある」ものがある．これらは，

明らかに満たす＝{1/スーザン, 1/リチャード, 1/スミス}

満たす可能性がある＝{1/レイモンド, 1/ビクター}

明らかに満たさない＝{1/トム}

となる． ■

質問の条件が「25 歳以上か」というようにファジィではないので，各集合に属するグレードは 0 か 1 である（0 のものは書いていない）．ファジィな条件をもつ質問をすると，上の各集合に属するグレードは 0 と 1 の間の値をとる．

また，関係中のデータが質問の条件を満たす可能性測度 (possibility measure) と必要性測度 (necessity measure) を用いて関係代数の演算を定義するという方法も提案されている[31)]．

さらに，ボーイング社で開発された RIM (Relational Information Management System) を基にした関係代数型の言語も実現されている[32)]．

④ 可能性分布-ファジィ関係モデル：③で述べたデータの値自身がもつあいまいさ以外に，データ間の関係についてのあいまいさもある．そこで，可能性分布-ファジィ関係モデルが提案されている[33)]．そして，このようなモデルに対して，関係代数と関係計算のほ

とんどの部分が定式化されている[33),34)].

⑤ 理論的取り扱い：最近，理論的な取り扱いとして，①のモデルに対するファジィ関数従属や無損失結合（lossless join）などの研究がある[35)].

以上，ファジィ・データベースについて紹介したが，データ・モデルを通常の関係モデルに近いレベルに設定すると，理論的な取り扱いやインプリメントが比較的容易になるが，現実世界のデータを表現する能力が足らなくなる．一方，現実世界のデータの表現に合わせてデータ・モデルを設定すると，理論的取り扱いやインプリメントが難しくなる．これらが適切に調和したデータ・モデルを見つける必要があるが，現在のところ，いろいろなデータ・モデルが提案されている段階である．また，関係モデル以外のデータ・モデルに対するファジィ・データベースも興味深い．

4. おわりに

以上，ファジィ集合とその演算についてまとめ，ファジィ集合の表現と処理，ファジィ推論，ファジィ・アルゴリズムとファジィ・プログラム，ファジィ・データベースについて概説した．内容が広い範囲にわたっているため，十分に説明できなかったものも多い．

人間の行っている情報処理は，ほとんどすべてがファジィ情報処理であるが，ファジィ情報処理の研究は，まだ始まったばかりである．今後の大いなる発展が期待される．

参考文献

1) Zadeh, L. A.: Fuzzy Sets, Inf. Control, Vol. 8, pp. 338-353 (1965).
2) Goguen, J. A.: L-Fuzzy Sets, Journal of Mathematical Analysis and Applications, Vol. 18, pp. 146-174 (1967).
3) Zadeh, L. A.: Quantitative Fuzzy Semantics, Inf. Sci., Vol. 3, pp. 159-176 (1971).
4) Zadeh, L. A.: The Concept of a Linguistic Variable and Its Application to Approximate Reasoning, Inf. Sci., Vol. 8, pp. 199-248; Vol. 8, pp. 301-357; Vol. 9, pp. 43-80 (1975).
5) 水本：最近のファジー理論，情報処理，Vol. 29, No. 1 pp. 11-22 (1988).
6) Zadeh, L. A.: A Fuzzy-Set-Theoretic Interpretation of Linguistic Hedges, J. Cybern., Vol. 2, pp. 4-34 (1971).
7) 馬野，水本，田中：Fuzzy 集合処理システムの構成，情報処理，Vol. 18, No. 9, pp. 884-892 (1977).
8) 馬野，久米：Lisp によるファジィ集合処理システムの概要，情報処理学会第35回（昭和62年後期）全国大会，pp. 789-790, No. 7Q-1 (1987).
9) 久米，馬野：Lisp によるファジィ集合処理システムの実現，情報処理学会第35回（昭和62年後期）全国大会，pp. 791-792, No. 7Q-2 (1987).
10) Davis, R., Buchanan, B. and Shortliffe, E. H.: Production Rules as a Representation for a Knowledge-Based Consultation Program, Artif. Intell., Vol. 8, pp. 15-45 (1977).
11) Winston, P. H.: Artificial Intelligence-Second Edition, Addison-Wesley (Reading, Mass., USA) (1984).
12) 廣田：ファジィ推論エキスパートシステムの現状の動向，情報処理，Vol. 28, No. 8 pp. 1065-1074 (1987).
13) 山川：ファジィコンピュータの原理と構造，Computer Today, No. 25, pp. 17-26 (1988).
14) 金井，石塚：Prolog-ELF：ファジィ論理を組み込んだ Prolog，情報処理学会論文誌，Vol. 27, No. 4, pp. 411-416 (1986).
15) Mukaidono, M., Shen, Z. and Ding, L.: Fuzzy Prolog, Second IFSA Congress, pp. 452-455 (1987).
16) Martin, T. P., Baldwin, J. F. and Pilsworth, B. W.: The Implementation of FPROLOG—A Fuzzy Prolog Interpreter, Fuzzy Sets and Systems, Vol. 23, pp. 119-129 (1987).
17) 馬野：ファジィ集合の概念を用いた FS-Prolog について，電子通信学会技術研究報告，Vol. 86, No. 345, pp. 87-95, COMP 86-81（コンピュテーション研究会）(1987).
18) Umano, M.: A Fuzzy Production System, in H. Prade and C. V. Negoita (eds.): Fuzzy Logic in Knowledge Engineering, Verlag TÜV Rheinland (Köln, West Germany), pp. 194-208 (1986).
19) 馬野：ファジィ・プロダクション・システムの実現，情報処理学会第32回（昭和61年前期）全国大会，pp. 1255-1256, No. 4M-11 (1986).
20) Zadeh, L. A.: Outline of a New Approach to the Analysis of Complex Systems and Decision Processes, IEEE Trans. on System, Man and Cybernetics, Vol. 3, pp. 28-44 (1973).
21) Tanaka, K. and Mizumoto, M.: Fuzzy Programs and Their Execution, in L. A. Zadeh, K. S. Fu, K. Tanaka and M. Shimura (eds.): Fuzzy Sets and Their Applications to Cognitive and Decision Processes, Academic Press (New York, USA), pp. 41-76 (1975).
22) Chang, C. L.: Interpretation and Execution of Fuzzy Programs, in L. A. Zadeh, K. S. Fu, K. Tanaka and M. Shimura (eds.): Fuzzy Sets and Their Applications to Cognitive and Decision Processes, Academic Press (New York, USA), pp. 191-218 (1975).
23) Tahani, V.: A Conceptual Framework for Fuzzy Query Processing—A Step toward Very

Intelligent Database Systems, Inf. Process. Manage., Vol. 13, pp. 289-303 (1977).
24) Zadeh, L. A.: PRUF—A Meaning Representation Language for Natural Languages, Int. J. Man-Mach. Stud., Vol. 10, pp. 395-460 (1978).
25) Baldwin, J. F.: FRIL—A Fuzzy Relational Inference Language, Fuzzy Sets and Systems, Vol. 14, pp. 155-174 (1984).
26) Zadeh, L. A.: Similarity Relations and Fuzzy Orderings, Inf. Sci., Vol. 3, pp. 177-200 (1971).
27) Buckles, B. and Petry, F.: A Fuzzy Model for Relational Databases, Fuzzy Sets and Systems, Vcl. 7, pp. 213-226 (1982).
28) Zadeh, L. A.: Fuzzy Sets as a Basis for a Theory of Possibility, Fuzzy Sets and Systems, Vol. 1, pp. 3-28 (1978).
29) 深海, 馬野, 水本, 田中: Fuzzy データベース検索操作言語について, 電子通信学会技術研究報告, Vol. 78, No. 233, pp. 65-72, AL 78-85 (オートマトンと言語研究会) (1979).
30) Umano, M.: FREEDOM-0: A Fuzzy Database System, in M. M. Gupta and E. Sanchez (eds.): Fuzzy Information and Decision Processes, North-Holland (Amsterdam, the Netherlands), pp. 339-347 (1982).
31) Prade, H. and Testemale, C.: Generalizing Database Relational Algebra for the Treatment of Incomplete or Uncertain Information and Vague Queries, Inf. Sci., Vol. 34, pp. 115-143 (1984).
32) Zemankova-Leech, M. and Kandel, A.: Implementing Imprecision in Information Systems, Inf. Sci., Vol. 37, pp. 107-141 (1985).
33) 馬野: Fuzzy 関係代数による fuzzy データベースからの検索, 電子通信学会技術研究報告, Vol. 81, No. 18, pp. 33-40, AL 81-15 (オートマトンと言語研究会) (1981).
34) 馬野, 深海, 水本, 田中: Fuzzy データベースからの検索処理について, 電子通信学会技術研究報告, Vol. 80, No. 204, pp. 45-54, AL 80-50 (オートマトンと言語研究会) (1980).
35) Raju, K. V. S. V. N. and Majumdar, A. K.: Fuzzy Functional Dependencies and Lossless Join Decomposition of Fuzzy Relational Database Systems, ACM Trans. Database Syst., Vol. 13, pp. 129-166 (1988).

(平成元年4月10日受付)