論 文

Fuzzy 集合処理システムの構成*

馬野元秀** 水本雅晴** 田中幸吉**

Abstract

This paper describes an implementation of a system for fuzzy sets manipulation which is based on FSTDS (Fuzzy-Set-Theoretić Data Structure), an extended version of childs' STDS (Set-Theoretic Data Structure) FSTDS language is considered as a fuzzy-set-theoretically oriented language which can deal with ordinary sets, fuzzy sets, fuzzy relations, L-fuzzy sets, level m fuzzy sets, type n fuzzy sets, and so on. The system consists of an interpreter, a collection of fuzzy set operations and the data structure, FSTDS, for representing fuzzy sets. FSTDS is made up of eight areas, namely, Fuzzy Set Area, Set Representation Area, Grade Area, Grade Tuple Area, Element Area, Element Tuple Area, Set Name Area and Set Operator Name Area.

FSTDS system, in which 52 fuzzy set operations are available, is implemented in FORTRAN, and is currently running on a FACOM 230-45S computer.

1. まえがき

現実の世界には，明確に定義することのできない，あいまいさを含んだものが数多く存在する．このようなあいまいさは以前から vagueness や ambiguity などとして研究されてきたが，1965年 L. A. Zadeh[1] により fuzzy 集合の概念が提案されて以来，盛んに研究され，種々の分野に応用されている．その応用分野としては，例えば，オートマトン理論，言語理論，論理，パターン認識，学習，意思決定，プログラム理論などをあげることができる[2]．

一方，通常の集合論の有用性は非常によく知られているものであり，集合を扱うことのできるシステムもいくつか作成されている．例えば，D. L. Childs の STDS[3),4)]，J. T. Schwartz の SETL[5]，片山らの LOREL[6] などがある．

STDS は FORTRAN などに埋め込んで使用でき，かなり豊富な集合演算を利用することができる．SETL は超高級言語の形をとり，例えば，$x+y$ で x, y に応じて，整数や実数の加算，集合の和集合，n 字組や文字列の連接を表せるが，演算は基本的なものしか存在しない．また，LOREL はデータ間の論理関係を問題とするようなシステムの処理のために作成されたものであり，通常の集合とは少し取り扱いが異なる．

これらの他にも，基本タイプとして集合をもつ PASCAL[7] や集合を基本とする言語である ABSET[8] などがある．

fuzzy 集合は，通常の集合の拡張と考えることができるので，fuzzy 集合を扱うことのできるシステムは，fuzzy 集合の応用の広さを考えると非常に大きな有用性をもつものと考えられる．

本論文では，Childs による STDS (Set-Theoretic Data Structure) の拡張として，fuzzy 集合や fuzzy 関係を記述・処理することのできるシステム (fuzzy 集合論的データ構造システム) の構成について述べ，いくつかの実行例を示す．なお，本 fuzzy 集合処理システムは FORTRAN によりインプリメントされ，FACOM 230-45S 上で稼動している．現在，52 種類の集合演算子が使用可能である．

2. Fuzzy 集合と Fuzzy 関係

後の準備のために，fuzzy 集合と fuzzy 関係について述べておこう．

* Implementation of a Fuzzy Sets Manipulation System by Motohide UMANO, Masaharu MIZUMOTO and Kokichi TANAKA (Faculty of Engineering Science, Osaka University)

** 大阪大学基礎工学部情報工学科

まず，fuzzy 集合は次のように定義される．

〔定義 1〕 集合 U における **fuzzy 集合** F とは，

$$\mu_F : U \rightarrow [0,1] \quad (1)$$

なるメンバーシップ関数 μ_F によって特徴づけられる集合である．値 $\mu_F(u) \in [0,1]$ は要素 $u \in U$ が fuzzy 集合 F に属する度合，すなわち**グレード**を表わす．

fuzzy 集合 F の表記法としては，

$$F = \{\mu_F(u_1)/u_1, \mu_F(u_2)/u_2, \cdots, \mu_F(u_n)/u_n\} \quad (2)$$

を採用する．ただし，$u_i \in U$ $(i=1,2,\cdots,n)$．

また，fuzzy 関係は次のように定義される．

〔定義 2〕 集合 U から集合 V への **fuzzy 関係** R とは，

$$\mu_R : U \times V \rightarrow [0,1] \quad (3)$$

なるメンバーシップ関数 μ_R によって特徴づけられる fuzzy 集合である．ここで，$U \times V$ は集合 U と集合 V の直積である．

fuzzy 関数 R の表記法としては，

$$R = \{\mu_R(u_1, v_1)/\langle u_1, v_1\rangle, \mu_R(u_1, v_2)/\langle u_1, v_2\rangle, \cdots, \mu_R(u_m, v_n)/\langle u_m, v_n\rangle\} \quad (4)$$

を用いる．ここで，$u_i \in U (i=1,2,\cdots,m)$，$v_j \in V$ $(j=1,2,\cdots,n)$ で，$\langle u_i, v_j\rangle$ は u_i と v_j の順序対を表す．

fuzzy 集合の概念が提案されて以来，いくつかの拡張が行われている．それらは，Goguen による L-fuzzy 集合[9]，Zedeh によるレベル m fuzzy 集合[10]，タイプ n fuzzy 集合[11]であり，それぞれ次のように定義される．

〔定義 3〕 集合 U における ***L*-fuzzy 集合** X，**レベル *m* fuzzy 集合** Y，**タイプ *n* fuzzy 集合** Z は，それぞれ次のようなメンバーシップ関数 μ_X, μ_Y, μ_Z により特徴づけられる集合のことである．

$$\mu_X : U \rightarrow L \quad (5)$$

$$\mu_Y : [0,1]^{[0,1]^{\cdots^{[0,1]^U}}} \rightarrow [0,1] \quad (6)$$

（$m-1$ 段）

$$\mu_Z : U \rightarrow [0,1]^{[0,1]^{\cdots^{[0,1]}}} \quad (7)$$

（n 段）

ここで，L は束を表す．

(注) レベル 1 fuzzy 集合，タイプ 1 fuzzy 集合は，通常の fuzzy 集合に対応する．

〔例 1〕 $U=\{a,b,c,d\}$ のとき，

$$F = \{0.1/a,\ 0.8/b,\ 0.9/d\} \quad (8)$$

は，U における fuzzy 集合であり，

$$R = \{0.3/\langle a,b\rangle,\ 0.9/\langle b,d\rangle\} \quad (9)$$

は，$U \times U$ における fuzzy 関係である．

〔例 2〕 $U=\{a,b,c,d\}$ のとき，

$$X = \{\langle 0.1, 0.9\rangle/a,\ \langle 0.8, 1\rangle/b,\ \langle 0.9, 0\rangle/c\} \quad (10)$$

は，U における L-fuzzy 集合となる*．また，

$$Y1 = \{0.3/a,\ 0.2/b,\ 0.9/d\} \quad (11)$$

$$Y2 = \{0.6/a,\ 0.1/b\} \quad (12)$$

のとき，

$$Y = \{0.6/Y1,\ 0.1/Y2\} \quad (13)$$

は，レベル 2 fuzzy 集合である．

$$high = \{1/1,\ 0.8/0.9,\ 0.4/0.8\} \quad (14)$$

$$middle = \{1/0.5,\ 0.5/0.6,\ 0.5/0.4\} \quad (15)$$

$$low = \{1/0,\ 0.8/0.1,\ 0.4/0.2\} \quad (16)$$

のとき，

$$Z = \{high/a,\ middle/b,\ low/d\} \quad (17)$$

は，タイプ 2 fuzzy 集合である．

3. Fuzzy 集合論的データ構造

fuzzy 集合や fuzzy 関係を処理するために，これらをどのようなデータ構造で表現するかを述べたのが，本章の fuzzy 集合論的データ構造 (FSTDS) で，D. L. Childs の STDS の拡張となっている[12),14)]．

FSTDS は，次の 8 個の領域から構成されている (**Fig. 1**, **Fig. 2**(次頁参照) を参照)．

①**F 集合領域** (FSA)：fuzzy 集合や fuzzy 関係を表すためのポインタの始点を格納するための領域で，各セルは各集合の内容を表わす集合表現領域へのポインタと，その集合の名前を表わす集合名領域へのポインタとの対よりなる．

②**集合表現領域** (SRA)：各集合の内容を表わす集合表現の集まりで，各集合表現は要素の数 n およびグレード部と要素部からなる．要素部は，通常の fuzzy 集

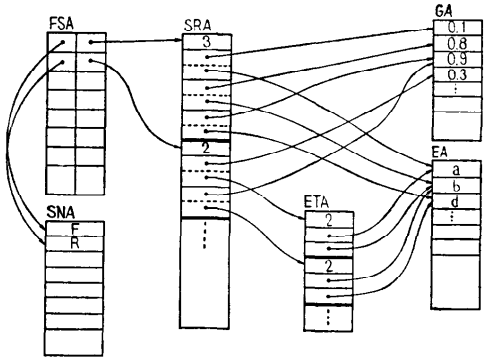

Fig. 1 Representation of a fuzzy set F anda fuzzy relation R by FSTDS

* ここで L は，$L=[0,1]\times[0,1]$ で順序関係が $\langle a,b\rangle \leq \langle c,d\rangle \leftrightarrow a \leq c$ かつ $b \leq d$ と定義される束である．本論文では，L-fuzzy 集合といえば，$L=[0,1]^n$ $(n=1,2,\cdots)$ の場合を意味する．

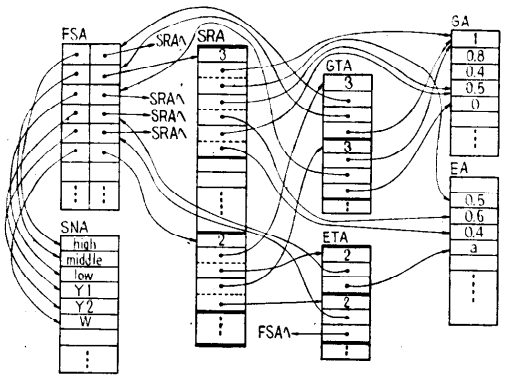

Fig. 2 Representation of a generalized fuzzy set W by FSTDS

合の場合には，要素領域へのn個のポインタよりなり，グレード部は要素部と対応して，グレード領域へのn個のポインタよりなる．さらに，要素部のポインタはfuzzy関係の場合には，要素組領域へのポインタとなり，レベルm fuzzy集合の場合には，F集合領域へのポインタとなる．また，グレード部のポインタは，L-fuzzy集合の場合には，グレード組領域へのポインタとなり，タイプn fuzzy集合の場合には，F集合領域へのポインタとなる．

③**グレード領域** (GA)：通常のグレードである区間[0,1]の数を格納するための領域．

④**グレード組領域** (GTA)：n字組のグレード（例えば，L-fuzzy集合のグレード）を格納する領域で，各n字組は整数nと，グレード領域，グレード組領域またはF集合領域のいずれかへのn個のポインタよりなる．

⑤**要素領域** (EA)：fuzzy集合の要素である文字列や数を格納する領域．

⑥**要素組領域** (ETA)：n字組の要素（例えば，n項fuzzy関係の要素）を格納する領域で，各n字組は整数nと，要素領域，要素組領域またはF集合領域のいずれかへのn個のポインタよりなる．

⑦**集合名領域** (SNA)：集合の名前を蓄えておく領域で，名前の部分とF集合領域へのポインタよりなる．

⑧**演算子名領域** (SONA)：集合演算子の名前を格納するための領域．

以上，8個の領域よりFSTDSは構成される．

〔**例 3**〕 例1で示したfuzzy集合Fとfuzzy関係Rは，FSTDSでは，Fig. 1のように示される．

例3で示したように，同じグレードの値や同じ要素は共有される．ただし，STDSで行われているような各集合の共通な部分集合を共有することは行わない[3]，これは，FSTDSの場合には，fuzzy集合を対象にするため，STDSのように簡単に互いに素な部分集合に分割することができないからである．

FSTDSにより，**2.** で述べたL-fuzzy集合，レベルm fuzzy集合，タイプn fuzzy集合を表現できるだけでなく，これらを組み合わせたさらに複雑なfuzzy集合をも表現することができる．

〔**例 4**〕 $U=\{a,b,c,d\}$のとき，例2における$Y1$，$Y2$および$high$, $middle$, lowが定義されているとすると，

$$W=\{\langle high, middle, 1\rangle/\langle Y1, a\rangle, \langle 1, low, 0\rangle/\langle Y2, Y1\rangle\} \quad (18)$$

というfuzzy集合（レベル2タイプ2 L-fuzzy関係）は，FSTDSではFig. 2のように表わすことができる．ただし，Fig. 2は紙面の都合で構成図の一部のみを示した．

FSTDSをFORTRANで表現する方法は，いくつか考えられるが，ここでは1つの整数型配列を一定の大きさのブロックにわけ，各領域には必要になったときに新しいブロックを割り付け，ブロック間はポインタにより結ぶという方法をとった．これは，各領域の再配置の回数やポインタ数の増加を防ぐだけでなく，仮想記憶の下ではページとブロックの大きさを調整することにより，効率よく処理することも考慮したからである．

以上で述べたように，fuzzy集合やfuzzy関係のデータ構造としてFSTDSを使用するが，ユーザがFSTDSの種々のポインタを管理し，操作するのは，非常に面倒であり，誤りも犯しやすい．そこで，これらを避けるためにポインタとは無関係にfuzzy集合やfuzzy関係を記述・処理できる手段をユーザに与える．これにより，ユーザはfuzzy集合やfuzzy関係が計算機上でどのように表現されているかということとは無関係に処理を行える．次章では，これについて述べる．

4. Fuzzy 集合論的データ構造システム

本章では，ユーザがFSTDSデータ構造*を利用す

* 以後，FSTDSという言葉は，単独では使用せずに，FSTDS語，FSTDSシステムのように使用する．今まで，FSTDSと呼んできたものはFSTDSデータ構造と呼ぶことにする．FSTDS語はユーザが記述するプログラム言語に対応し，それを処理するのがFSTDSシステムで，FSTDSシステムが保持しているデータ構造がFSTDSデータ構造である．

る場合の表現方法である FSTDS 語とそれを処理する FSTDS システム[12),13)]について述べる.

FSTDS 語による記述は，できるだけユーザに解りやすい表現法にした．すなわち，

$$opr(opd1, opd2, \cdots, opdn) \qquad (19)$$

の形で表わす．ここで，opr は集合演算子で $opdi$ $(i=1,2,\cdots,n)$ は被演算子であり，opr に応じて集合であったりグレード／要素の組であったりする．また，演算子は任意に入れ子にできる．

演算については，ユーザが複雑なグレードの計算を行う必要がないように，できるだけシステムの方で行うことにした．したがって，ユーザは fuzzy 集合を定義し，それをどのように処理するかを書くだけでよい．現在，使用可能な集合演算子は 52 種類である．これについては，次章で述べることにして，本章では FSTDS 語による記述例を示す．

〔例 5〕 FSTDS 語では，**Fig. 3** のように書くと，Uは通常の集合，F は fuzzy 集合，R は fuzzy 関係を表わすことになる．

```
U:=SET(A,B,C,D);
F:=FSET(0.8/A, 0.9/B, 1/C, 0.1/D);
R:=FSET(0.1/<A,B>, 0.9/<C,D>);
```

CPU 時間*: 203 msec

Fig. 3 An ordinary set U, a fuzzy set F and a fuzzy relation R in FSTDS language

〔例 6〕 **Fig. 4** のような fuzzy 有向グラフは，FSTDS 語では **Fig. 5** のように記述することができる．ここで，V は頂点の集合，A は有向辺の fuzzy 集合を表わし，G が fuzzy 有向グラフを表わす．

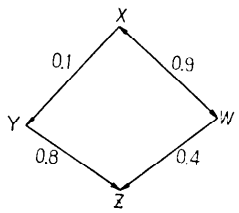

Fig. 4 Fuzzy directed graph G

```
V:=SET(X,Y,Z,W);
A:=FSET(0.1/<X,Y>, 0.8/<Y,Z>, 0.4/<W,Z>,
                   0.9/<X,W>, 0.9/<W,X>);
G:=SET(<V,A>);
```

CPU 時間: 266 msec

Fig. 5 FSTDS language statements for fuzzy directed graph Gin Fig. 4

〔例 7〕 X, Y を fuzzy 集合として，R を fuzzy 関係とすると，

$$(X \cup Y) \circ R = (X \circ R) \cup (Y \circ R) \qquad (22)$$

が成立する．ここで，∪は和集合の演算を表わし，∘は $X \circ R$ で R による X の像を表わす．いま，

$$X = \{1/a,\ 0.9/b,\ 0.3/c\}$$

$$Y = \{0.1/a,\ 0.7/b,\ 0.9/c\}$$

$$R = \begin{bmatrix} 1 & 0.8 & 0 \\ 0.7 & 1 & 0.2 \\ 0 & 0.5 & 0.1 \end{bmatrix}$$

の場合に，FSTDS 語では，**Fig. 6**(a)(次頁参照)のように記述することができ，実行結果は Fig. 6(b)のようになる．また，各行の実行時間は **Fig. 6**(c)，各集合演算子の実行回数と実行時間は，**Fig. 6**(d)のようになる．

〔例 8〕 **Fig. 7** のような fuzzy 知識は FSTDS 語では，**Fig. 8**(a)のようなタイプ 2 レベル 2 fuzzy 集合 ANIMAL で記述できる．さらに，BAT がどのような集合に属しているかは，FSTDS 語では，Fig. 8(b)のように書くと，処理結果として Fig. 8(c)が得られる．

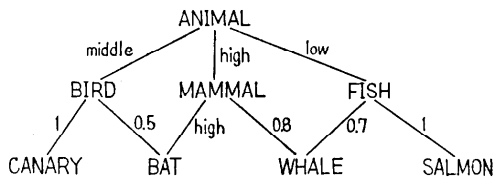

Fig. 7 An example of fuzzy knowledge

```
LOW:=FSET(1/0, 0.8/0.1, 0.3/0.2);
MIDDLE:=FSET(0.6/0.6, 1/0.5, 0.6/0.4);
HIGH:=FSET(1/1, 0.8/0.9, 0.3/0.8);
BIRD:=FSET(1/CANARY, 0.5/BAT);
MAMMAL:=FSET(HIGH/BAT, 0.8/WHALE);
FISH:=FSET(0.7/WHALE, 1/SALMON);
ANIMAL:=FSET(MIDDLE/BIRD, HIGH/MAMMAL, LOW/FISH);
```

(a) Fuzzy knowledge

```
ISA:=CONVERSE(RELATION(ANIMAL));
X:=IMAGE(ISA,SET(BAT));
PRINT(X);
```

(b) Manipulation

```
FSET(0.5/BIRD, HIGH/MAMMAL);
```

(c) Output

CPU時間:1082 msec

Fig. 8 Fuzzy knowledge and its manipulation by FSTDS languae (a)

以上の例 5～例 8 からもわかるように FSTDS 語は次のような構文規則に従う．

(1) 形式は自由形式であり，文の終りは；で示

* ここでいう CPU 時間は，システムの初期化などの時間を除いたものである．システムの初期化などに約 2 秒を要する．以下の例における CPU 時間も同様である．

```
FSTDS SYSTEM SOURCE LIST

1       PARA(A=1,T=1);
2       X:=FSET(1/A, 0.9/B, 0.3/C);
3       Y:=FSET(0.1/A, 0.7/B, 0.9/C);
4       R:=FSET(1/<A,A>,0.8/<A,B>,0.7/<B,A>,1/<B,B>,
5                 0.2/<B,C>,0.5/<C,B>,0.1/<C,C>);
6       PRINT(ASSIGN(Z,UNION(X,Y)));
7       PRINT(IMAGE(R,Z));
8       V:=IMAGE(R,X);  W:=IMAGE(R,Y);
9       PRINT(UNION(V,W));
10      END;
```

(a) Program

```
FSET(1/A, 0.9/B, 0.9/C);    ------ XUY
FSET(1/A, 0.9/B, 0.2/C);    ------ (XUY)・R
FSET(1/A, 0.9/B, 0.2/C);    ------ (X・R)U(Y・R)
```

(b) Output

CPU時間：606msec

```
1.  37MSEC
2.  45MSEC
3.  48MSEC
4.  66MSEC
5.  57MSEC
6.  55MSEC
7.  51MSEC
8.  62MSEC
9.  46MSEC
```

FSTDS文の1行の実行時間
これは，カードを一枚読んだ時
に時間を計算するもので，
PARA文の T=1 で時間が出
力されるようになる．したがって
オーバーヘッドなどの時間もすべて
含む

(c) CPU twe for each laie

```
****** NORMAL END ..... FSTDS SYSTEM ******


GARBAGE COLLECTION 1 .....  0 TIMES
GARBAGE COLLECTION 2 .....  0 TIMES
GARBAGE COLLECTION 3 .....  0 TIMES

THE NUMBER OF EXECUTED STATEMENTS .........   10
THE NUMBER OF EXECUTED SET OPERATIONS .....   24

  NO SET OPERATION          COUNT      MSEC  MSEC/COUNT
   1 SET ..................     3        10        3.33  ←これは，システムで定義せずに
   2 FSET .................     3       144       48.00    使用できるEMPTY,TRUE,FALSEも
   3 DUMP .................     0         0              システムが定義するときのもの．
   4 SNAP .................     0         0
   5 PARA .................     1         9        9.00
   6 PRINTL ...............     0         0
   7 ELEMENT ..............     0         0
   8 SOP ..................     0         0
   9 CUT ..................     0         0
  10 EXP ..................     0         0
  11 ASSIGN ...............     9         6        0.67
  12 UNION ................     2         2        1.00
  13 UNIONA ...............     0         0
  14 INT ..................     0         0
  15 INTA .................     0         0
  16 PROD .................     0         0
  17 PRODA ................     0         0
  18 ASUM .................     0         0
  19 ASUMA ................     0         0
  20 ADIF .................     0         0
  21 ADIFA ................     0         0
  22 BSUM .................     0         0
  23 BSUMA ................     0         0
  24 BDIF .................     0         0
  25 BDIFA ................     0         0
  26 PRINT ................     3        21        7.00
  27 DOMAIN ...............     0         0
  28 RANGE ................     0         0
  29 CONVERSE .............     0         0
  30 COMPOSE ..............     0         0
  31 CP ...................     0         0
  32 RS ...................     0         0
  33 IMAGE ................     3        13        4.33
  34 CIMAGE ...............     0         0
  35 NORM .................     0         0
  36 CON ..................     0         0
  37 DIL ..................     0         0
  38 CINT .................     0         0
  39 EQ ...................     0         0
  40 SUBSET ...............     0         0
  41 DISJOINT .............     0         0
  42 CD ...................     0         0
  43 # ....................     0         0
  44 MAXG .................     0         0
  45 RELATION .............     0         0
  46 SF ...................     0         0
  47 GF ...................     0         0
  48 PRINTB ...............     0         0
  49 PRINTS ...............     0         0
  50 PRINTN ...............     0         0
  51 DLT ..................     0         0
```

(d) List of the count and CPU time of tuzzy set operators

Fig. 6 Program and output of $(X \cup Y) \circ R = (X \circ R) \cup (Y \circ R)$

す．したがって，1行に複数個の文が書けるし，1つの文が複数行にわたってもよい．このとき，継続を示す記号などは必要としない．

(2) 空白は一切無視される．

(3) 演算子は何重にでも入れ子にできる．

(4) 文字列は，それぞれの演算子ごとに規則を設けて処理し，通常のプログラム言語で行われているような ' で括ることは行わない．

(5) 式はすべて値をもつ．文も同様である．そして，その値は常に fuzzy 集合である．

(6) プログラムは END 文で終了する．

FSTDS 語の構文を超言語を用いて定式化すること

もできる[4]が，構文が簡単でもあるのでここでは省略する．

FSTDS 語には例からもわかるように，ラベルも制御文も存在しないので，文を順に実行して処理を終る．制御文を利用したい場合には，FORTRAN との結合を行い FORTRAN の制御文を利用する．

これについて，例を示そう．

〔例 9〕 FSTDS 語と FORTRAN により，**Fig. 9** (a) のように書くと，実行結果は Fig. 9 (b) のようになる．このように FORTRAN の中に FSTDS 語の文を書くことができる．このとき，FORTRAN と FSTDS 語の文を区別するために FSTDS 語の文の前にはFをつける．Fig. 9 (a) は，3で X と U を空集合にし，4~8 の DO ループで X と U に要素をつけ加えて X と U を定義し，9で U と X を出力し，10で U と X の絶対差集合[1] (X の補集合に対応する) を求めて出力し，11で停止するというプログラムである．

```
                FSTDS SYSTEM SOURCE LIST

 1   F      PARA(G=1)
 2          N=5
 3   F      X=U=EMPTY
 4          DO 10 I=1,N
 5          G=FLOAT(I)/FLOAT(N)
 6   F      X=UNION(X,FSET(!!G/!!))
 7   F      U=UNION(U,SET(I))
 8       10 CONTINUE
 9   F      PRINT(U,X)
10   F      PRINT(ADIF(U,X))
11          STOP
12          END
```

(a) FORTRAN program with FSTDS language

```
U=FSET(1.0/1, 1.0/2, 1.0/3, 1.0/4, 1.0/5);
X=FSET(0.2/1, 0.4/2, 0.6/3, 0.8/4, 1.0/5);
FSET(0.8/1, 0.6/2, 0.4/3, 0.2/4);
```

(b) Output

CPU時間:343msec

Fig. 9 FSTDS language embeded in FORTRAN

このようなプログラムは FSTDS トランスレータにより，FORTRAN だけからなるプログラムに変換される．すなわち，FSTDS 語の文はいくつかの CALL 文に展開され，FORTRAN コンパイラで翻訳され，結合編集時に必要な SUBROUTINE が結合され実行に移される．

FORTRAN との結合のための FSTDS 語の変更としては，FORTRAN の変数を FSTDS 語内で利用するために FORTRAN の変数に！または!!をそれぞれ整変数か実変数かに応じて前につけること，行の終りを文の終りとするので2行以上の文には継続の印を必要とすること，:=は=でもよいことなどがあげられる．

FSTDS 語における fuzzy 集合の内容を FORTRAN の側で利用する場合には，そのために準備された特定の SUBROUTINE を CALL すればよい．これについての詳細は省略する．

このようにして，FSTDS 語と FORTRAN を結合して使用することにより，FSTDS 語の機能を大幅に拡大することができる．このことは逆にみると，FORTRAN に fuzzy 集合を取り扱う機能を与えたことになり，FORTRAN の応用範囲が広がるものと考えられる．

5. FSTDS システムの集合演算子

FSTDS システムには，現在，52種類の集合演算子が準備されており，ユーザはこれらを組み合わせて目的を果たすことができる．

これらの演算子を **Table 1** (次頁参照) に示す．なお，fuzzy 集合に対するこれらの演算がどのように定義されるかは文献[1),2),9),10),11),14),15)]を参照されたい．

これらの集合演算子は次の8種類に分類される．

① **集合構成演算子**：その被演算子を要素やグレードとする集合，fuzzy 集合，fuzzy 関係などを構成する演算子で，SET が通常の集合や関係を，FSET が fuzzy 集合や fuzzy 関係を構成するのに使用される．

② **代入演算子**：集合名に集合を割り付けるときに使用する．:=と ASSIGN があり，:=は多重代入が可能であるが文のトップ・レベルでしか使用できない．ASSIGN は，多重代入はできないが式の中で使用できる．これらの代入演算は，ポインタのつけかえでなく，その集合表現をすべてコピーしているので，まったく独立に集合を更新・再定義することができる．

③ **Fuzzy集合間の演算を行う演算子**：2つ以上の fuzzy 集合の演算を行うもので，UNION, INTERSECTION, PROD, ASUM, ADIF, BSUM, BDIF がある．これらは，被演算子である fuzzy 集合で同じ要素のグレードに種々の演算を行うものである．例えば，UNION では同じ要素のグレードの max をとる．そのとき，存在しないものはグレードを0と考えて演算を行う．引数の数は $n \geq 2$ であるが，3以上の場合には前から順に2項演算を行い，その結果とその次のものとの2項演算を行うことにより実行される．さらに，これらの演算子名のうしろにAのついた演算子があり (例えば，UNIONA)，これらは被演算子に fuzzy 集合の集合を1個とり，その要素である fuzzy 集合にわたってその演算 (例えば，UNION) を行う．

Table 1 Fuzzy set operations available in FSTDS system

集合演算子	引数	機能	集合演算子	引数	機能
SET($u_1, u_2, \cdots, u_n$)	$n\geq 0$	通常の集合を構成する	EQ(X_1, X_2)	2	X_1 と X_2 は等しいか
FSET($\mu_1/u_1, \cdots, \mu_n/u_n$)	$n\geq 0$	fuzzy 集合を構成する	SUBSET(X_1, X_2)	2	X_1 は X_2 の部分集合か
ASSIGN(Y, X)	2	$Y=X$ と同じ	DISJOINT($X_1, \cdots, X_n$)	$n\geq 2$	$X_1, X_2, \cdots, X_n$ は互いに素か
UNION($X_1, X_2, \cdots, X_n$)	$n\geq 2$	$X_1, X_2, \cdots, X_n$ の和集合	ELEMENT($\mu/u, X$)	2	μ/u が X に存在するか
INTERSECTION ($X_1, X_2, \cdots, X_n$)	$n\geq 2$	$X_1, X_2, \cdots, X_n$ の共通集合	CUT($\mu_1/\mu_2, X$)	2	X を μ_2 でカットし，グレードを μ_1 にする
PROD($X_1, X_2, \cdots, X_n$)	$n\geq 2$	$X_1, X_2, \cdots, X_n$ の代数積集合	SOP($\mu/n, X$)	2	n に応じ，グレード μ と X の種々の演算
ASUM($X_1, X_2, \cdots, X_n$)	$n\geq 2$	$X_1, X_2, \cdots, X_n$ の代数和集合	EXP($\mu/x, X$)	2	$\mu \wedge X^x$ (X の x 乗)
ADIF($X_1, X_2, \cdots, X_n$)	$n\geq 2$	$X_1, X_2, \cdots, X_n$ の絶対差集合	DIL(X)	1	$X^{0.5}$
BSUM($X_1, X_2, \cdots, X_n$)	$n\geq 2$	$X_1, X_2, \cdots, X_n$ の限界和集合	CON(X)	1	X^2
BDIF($X_1, X_2, \cdots, X_n$)	$n\geq 2$	$X_1, X_2, \cdots, X_n$ の限界差集合	CINT(X)	1	contrast intensification
UNIONA(X)	1		NORM(X)	1	X を正規化
INTERSECTIONA(X)	1		CD(X)	1	X のグレードの和
PRODA(X)	1		$\sharp$ (X)	1	X の要素の数
ASUMA(X)	1	X にわたって，それぞれの演算を行う	MAXG(X)	1	X の最大グレード
ADIFA(X)	1		SF(X, K)	2	K による X の support fuzzification
BSUMA(X)	1		GF(X, K)	2	K による X の grade fuzzification
BDIFA(X)	1		DLT($X_1, X_2, \cdots, X_n$)	$n\geq 1$	$X_1, X_2, \cdots, X_n$ をシステムから消去
COMPOSE($R_1, \cdots, R_n$)	$n\geq 2$	関係 $R_1, R_2, \cdots, R_n$ の合成	PRINT($X_1, X_2, \cdots, X_n$)	$n\geq 1$	$X_1, X_2, \cdots, X_n$ を出力
CONVERSE(R)	1	関係 R の逆関係	PRINTB($X_1, X_2, \cdots, X_n$)	$n\geq 1$	Boolean 型で $X_1, \cdots, X_n$ を出力
IMAGE(R, X)	2	X の R による像	PRINTS($X_1, X_2, \cdots, X_n$)	$n\geq 1$	通常の集合の型で $X_1, X_2 \cdots, X_n$ を出力
CIMAGE(R, X)	2	X の R による逆像	PRINTN($X_1, X_2, \cdots, X_n$)	$n\geq 1$	名前つきで $X_1, X_2, \cdots, X_n$ を出力
DOMAIN(R)	1	関係 R の定義域	PRINTC (文字列)	—	文字列を出力．／は改行
RANGE(R)	1	関係 R の値域	DUMP($\alpha_1, \alpha_2, \cdots, \alpha_n$)	$n\geq 1$	領域をダンプする
CP($X_1, X_2, \cdots, X_n$)	$n\geq 2$	直積 $X_1\times X_2\times\cdots\times X_n$	SNAP(α)	1	定義されているすべての集合を出力
RS(R, X)	2	R の X による制限	PARA ($\alpha_1=\beta_1, \cdots, \alpha_n=\beta_n$)	$n\geq 1$	各種オプションの指示
RELATION(X)	1	レベル m fuzzy 集合 X を関係に変換する	END(X)	0 か 1	プログラムの実行をおわる

(注) 1. 記号の意味は次のとおり，$u_1, u_2, \cdots, \mu_1, \mu_2, \cdots$，なども同様に使用する．
u: 要素に対応し，数，文字列またはすでに定義された集合，またはそれらの n 字組．
μ: グレードに対応し，区間 $[0, 1]$ の数またはすでに定義された集合，またはそれらの n 字組．
X: すでに定義された fuzzy 集合または式．
Y: すでに定義された fuzzy 集合または新しく定義される集合名．
R: すでに定義された fuzzy 関係．
$\mathcal{X}$: すでに定義された fuzzy 集合の集合．
n: 整数
x: 実数．
α: 英字1文字．
K: fuzzy 化するときに使用される核集合と呼ばれる fuzzy 集合の集合
β: PARA のオプション．
2. SET, FSET で $n=0$ のときは，空集合を意味する．
3. END(X) は，単に END( ) または END としてもよい．
4. fuzzy 集合に関する演算については，文献[1),2),8),10),11),14),18)]を参照されたい．

④ **Fuzzy 関係に関する演算子**: fuzzy 関係は FSET を使用して，n 字組を要素とする fuzzy 集合として定義できる．fuzzy 関係を fuzzy 集合とみて和集合や共通集合を求めることは，③の演算子を使用することにより可能である．fuzzy 関係に関する演算子は，n 字組の各要素に対する演算を行うものであり，COMPOSE, CONVERSE, IMAGE, CIMAGE, DOMAIN, RANGE, CP, RS, RELATION がある．

⑤ **真理値上の集合を値にもつ演算子**: 通常のプログラミング言語では，真または偽という値をもつ演算子に対応する．FSTDS 語では，結果の値はすべて fuzzy 集合となるので，これらは真理値の上での fuzzy 集合を値としてもたせることにする．すなわち，真に対しては FSET (1/1) が対応し，偽に対しては FSET (1/0) が対応する[11)]．しかし，これらの集合を TRUE, FALSE という形で出力する演算子 (PRINTB) が用意されているのでユーザは出力時にはこれを使用すればよい．真理値上の集合を値にもつ演算子には，EQ, SUBSET, DISJOINT, ELEMENT の4つがある．

⑥ **Fuzzy 集合に対する他の演算子**: fuzzy 集合と fuzzy 集合の間の演算を行うものではなく，1つの fuzzy 集合に対して種々の演算を行う演算子が主であり，α レベル集合を求めるもの (CUT)，グレードに種々の演算を行うもの (SOP, EXP, DIL, CON, CINT, NORM)，グレードの情報などを得るもの (CD, $\sharp$, MAXG)，fuzzy 化を行うもの (SF, GF)，システムか

らその集合を消去するもの (DLT) がある.

⑦ **出力演算子**: 通常の集合や関係, fuzzy 集合や fuzzy 関係あるいはより一般的な fuzzy 集合を出力するものと文字列を出力するものとからなる. 任意の型の fuzzy 集合や fuzzy 関係を出力する PRINT, ⑤で述べた真理値上の2つの特別な fuzzy 集合を TRUE, FALSE と出力する PRINTB, 通常の集合や関係を出力するための PRINTS, 集合名を前につけて出力する PRINTN, 文字列を出力する PRINTC がある.

⑧ **デバッグまたは出力制御のための演算子**: デバッグ用には, そのときの FSTDS データ構造を出力する DUMP とそのときまでに定義された集合をすべて名前付きで出力する SNAP がある. また, デバッグや出力制御などのために種々の情報をシステムに与える PARA がある.

現在, これらの演算子は FORTRAN の SUBROUTINE の形でインプリメントされている.

6. むすび

FSTDS システムは, FSTDS データ構造, それを取り扱う fuzzy 集合演算, さらに, FSTDS 語のプログラムを解析して fuzzy 集合演算を呼び出すインタプリタからなっている. これらを, FACOM 230-45S OS II/VS 上の FORTRAN S でインプリメントした. 4. における FSTDS 語やその処理結果の例はこれによるものである. また, FORTRAN との結合も FSTDS トランスレータを作成したので可能である. FSTDS システムは, 現在, 116 k バイトを必要とする. これは, インタプリタ形式のためシステムの過半をしめる fuzzy 集合演算の SUBROUTINE がすべて結合されているからである. これについては, FSTDS 語の文を直接集合演算の列に変換するコンパイラを作成することにより, 使用しない集合演算は結合せずに処理できるので, 使用メモリをかなり小さくできると思われる. 処理時間は, 例5～例9で述べたように要素が数個よりなる fuzzy 集合を構成し, 代入するのに 60～70 msec を要する. これは FORTRAN で文字処理を行っているのに加えて, 同一の要素やグレードを共有するためにグレード領域や要素領域などを探索しなければならないし, その際, FSTDS データ構造が FORTRAN の配列上に作成されているので毎回アドレス計算を行わなければならないからである. しかし, FORTRAN S の CPU 時間を測定する CLOCKM という基本 SUBROUTINE を使用した結果*, fuzzy 集合の構成 (FSET や SET) に比べるとポインタの操作だけを行う他の演算はかなり速いことがわかった. 処理時間は, アセンブラを使用して FSTDS データ構造を直接参照できるようにするだけでも, かなり改良されると思われる.

機能的には, L-fuzzy 集合やタイプ n fuzzy 集合に関しては一部の演算しか行えないが, 演算方法については, Zadeh や Goguen などにより定式化されている[9),11)]ので, 現在, これらを処理できるように拡張しているところである. これについては, 別の機会に述べることにする.

FSTDS システムでは, FSTDS 語により fuzzy 集合のレベルで問題を記述できるので, 大きなシステムを作成するときに fuzzy 集合の表現法やグレードの計算に注意を払う必要もないし, FORTRAN からも呼べるので, かなり細かい複雑な処理を行うことができる. しかし, FSTDS 語の記述で式19) のような prefix 形の表記法だけではわかりにくいので, 通常の集合論などで行われているような infix 形の表記法の導入も考えている. また, fuzzy 集合の表現法として FSTDS データ構造を使用したがより適した表現法の研究やより効率のよい処理方法の研究も大切であろう.

現在, FSTDS システムは fuzzy 推論を行うシステムの作成に応用されている. この他にも, fuzzy 集合の適用できる数多くの分野で使用することができるであろう.

謝辞 本研究を遂行するにあたり有益な御助言をくださいました本学の豊田順一助教授に深く感謝します. また, システム作成に協力くださった大崎浩一氏 (現在, 住友海上火災) に感謝します.

* PARA 文の指定により, 各演算子ごとに使用回数, CPU 時間, 平均 CPU 時間の表を END 文の実行後に出力できる (Fig. 6 参照).

参考文献

1) L. A. Zadeh: Fuzzy Sets, Information and Control, Vol. 8, pp. 338～353 (1965).
2) L. A. Zadeh, K. Tanaka et al. (eds.): Fuzzy Sets and Their Applications to Cognitive and Decision Processes, Academic Press Inc., New York (1975).
3) D. L. Childs: Description of a Set-Theoretic Data Structure, FJCC, pp. 557～564 (1968).
4) D. L. Childs: Feasibility of a Set-Theoretic Data Structure—A General Structure Based on a Reconstituted Definition of Relation, Informa-

tion Processing 68, pp. 420～430 (1968).
5) J. T. Schwartz: Automatic Data Structure Choice in a Language of Very High Level, Comm. ACM, Vol. 18, pp. 722～728 (1975).
6) 片山卓也, 日比野靖他: 論理関係処理言語 LOREL-1, 情報処理, Vol. 15, pp. 326～334 (1974).
7) N. Wirth: The Programming Language Pascal, Acta Informatica, Vol. 1, pp. 35～63 (1971).
8) E. W. Elcock, J. M. Foster et al.: Abset, A Programming Language Based on Sets: Motivation and Examples, in B. Meltzer and D. Michie (eds.), Machine Intelligence 6, pp. 467～492, Edinburgh University Press (1971).
9) J. A. Goguen: L-Fuzzy Sets, J. Math. Anal. Appl., Vol. 18, pp. 145～174 (1967).
10) L. A. Zadeh: Quantitative Fuzzy Semantics, Information Sciences, Vol. 3, pp. 159～176 (1971).
11) L. A. Zadeh: The Concept of a Linguistic Variable and its Application to Approximate Reasoning I, II, III, Information Sciences, Vol. 8, pp. 199～250, pp. 301～358, Vol. 9, pp. 43～80 (1975).
12) 馬野, 水本, 豊田, 田中: Fuzzy 集合論的データ構造システム作成計画, 情報処理学会第 16 回大会 199 (1975).
13) 馬野, 大崎, 水本, 豊田, 田中: Fuzzy 集合論的データ構造システムについて, 昭和 51 年度電子通信学会総合全国大会 1101 (1976).
14) 馬野: Fuzzy 集合論的データ構造システムの構成, 大阪大学修士論文 (1976).
15) L. A. Zadeh: A Fuzzy-Set-Theoretic Interpretation of Linguistic Hedges, Journal of Cybernetics, Vol. 2, pp. 4～34 (1972).

(昭和 51 年 7 月 27 日受付)
(昭和 51 年 11 月 30 日再受付)