

形式仕様記述言語 VDM++ に出力結果を考慮した ファジィ集合を作成する手法と推論の提案

A Method for Creating Fuzzy Sets and Its Inference Considering Output Results in Formal Specification Description Language VDM++

牧田 蒼斗 † 和崎 克己 ††

Aoto Makita Katsumi Wasaki

1 はじめに

ソフトウェア開発の上流工程において、形式手法を用いてシステムの仕様の妥当性確認をすることで、下流工程での手戻りコストの削減やシステムの高品質化につながる。しかし、自然言語によって書かれた仕様には曖昧さが存在することがあり、それによって開発時に、仕様の認識の齟齬で、システムに不備が生じることも少なくない。そのため、形式手法を用いて妥当性確認を行う際に、この曖昧さのうち数値として評価できるものを表現する必要がある。

本研究では、形式手法の一つである仕様記述言語 VDM++ において、ファジィ理論を用いて曖昧さを数値として表現し、評価することが目的である。

2 形式仕様記述言語 VDM++ の概要

形式手法の一つである VDM++ は、不変条件・事前条件・事後条件・陰仕様記述などの形式手法を支援する機能だけでなく、クラス・多重継承といったオブジェクト指向プログラミングおよび、パターン・内包表記・型変数・高階関数などの関数型プログラミング機能も持っている。VDM++ で科学的・系統的な分析や検証を行うことにより、誤りを早期に発見し手戻りを防いだり、品質を高めることができる [1]。

しかし、以前の研究 [2] で、サイバーフィジカルシステムの妥当性確認を行う際に、仕様の曖昧な表現が検証に組み込めない事象が存在した。VDM++ は、数学的・論理的に検証を行うため、曖昧な表現を数値化する必要がある。そこで本研究では、仕様の曖昧な部分のうち表現可能な部分に着目し、数値として評価することを目指す。また、VDM++ の機能を用いて、数値化する際のプロセスの妥当性確認を行う。

3 ファジィ理論

3.1 ファジィ理論

ファジィ理論とは、ファジィネスという不確かさを扱う理論枠組のことであり、ファジィ集合論、ファジィ論理、ファジィ測度論の総称である。ファジィ論理というのは多値論理にファジィ集合概念を導入したもので、これ自体独自の理論ではないが、他のどの論理とも異なる特徴をもっており、制御や人工知能へファジィ理論の応用に際し、とくに重要なものである [3]。

ファジィ理論は特に、人間の意思決定部分に応用されていることが多い。浄水場の応用例 [4] では、水道の衛生的安全性を守るための塩素の注入は、日射などの環境に左右され、熟練したオペレータの監視・介入を必要とする。そこで、ファジィ理論を用い、浄水場で働くオペレータの操作モデルを作ることにより、制御プロセスの自動化を実現している。このように、ファジィ理論は、物理空間内における曖昧性判断を行う際に効果的である。

3.2 ファジィ集合

ファジィ集合とは、明確に表せない曖昧さの程度を取り入れ表現したもので、曖昧な表現を数学的に取り扱うために使用される。また、ファジィ集合を数学的に厳密に取り扱うために、各要素毎にグレードという数値を導入する。グレードは、0 と 1 の間の実数値であり、1 に近いほど、その要素は集合に属している度合いが強いことを表す。度合いの関数をメンバーシップ関数と呼び、曖昧な値を数量化し間接的に定義することができる。

3.3 グレード写像群

本研究では、1 つ 1 つのメンバーシップ関数の集合をメンバーシップ関数コレクションと定義し、値を入れることでグレードのリストを返す、グレード写像群を VDM++ 記述で作成した。本論文では、三角型メンバーシップ関数を例に挙げる。

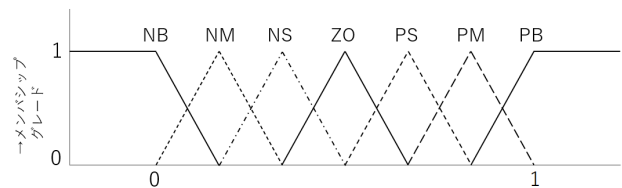


図 1 メンバーシップ関数コレクション

$$\mu(x) = \begin{cases} \frac{2x}{width} - \frac{2top}{width} + 1 & (top - \frac{width}{2} \leq x < top) \\ -\frac{2x}{width} + \frac{2top}{width} + 1 & (top \leq x < top + \frac{width}{2}) \\ 0 & (x < top - \frac{width}{2}, top \leq x) \end{cases} \quad (1)$$

メンバーシップ関数は、事前にはっきりと定義することは困難である。そのため、パラメータ化して定義しておき、数値を与えてパラメータを決定する方法をとった。

式 (1) は、三角型メンバーシップ関数を作成する際に用いた式である。top, width はそれぞれ三角型メンバー

† 信州大学大学院総合理工学研究科, Graduate School of Science and Technology, Shinshu University

†† 信州大学工学部電子情報システム工学科, Department of Electrical and Computer Engineering, Faculty of Engineering, Shinshu University

シップ関数の三角形の頂点と底辺の幅を表しており、この数値を変えることで様々な形の三角型メンバーシップ関数が作成可能である。

しかし、メンバーシップ関数の性質を満たさない関数が出てきてしまう可能性があるため、VDM++ の機能を使って関数が制約条件を満たすか検証する必要がある。

4 メンバーシップ関数の記述

次に VDM++ 記述例を示す。以下は、式 (1) のメンバーシップ関数の記述部分である。

三角型関数の頂点と底辺の幅の列を与え、求めたいグレードの要素 x を設定することで、グレード値を得ることができる。また、事前条件では、引数の列が正しいこと、事後条件では、グレードが 0 以上 1 以下の数であるかを確認している。

```
operations
public MakeTri: seq of real ==> real
MakeTri(list) ==
  let top = list(1),
      width = list(2) in
  (dcl grade: real := 0;
   dcl x: real;
   x := 1;
   if top - width/2 <= x and x < top then
     grade := 2*x/width - 2*top/width + 1
   elseif top <= x and x < top + width/2 then
     grade := -2*x/width + 2*top/width + 1
   elseif x <= 0 then
     grade := 1
   elseif 1 <= x then
     grade := 1
   else
     grade := 0;
   return grade)
pre len list > 1
post RESULT >= 0 and RESULT <= 1;
```

Listing 1 VDM++ メンバーシップ関数記述

5 メンバーシップ関数のテスト

5.1 メンバーシップ関数の妥当性

パラメータとして定義したメンバーシップ関数の妥当性確認を行うために、制約条件を設けてテストをする必要がある。はじめにメンバーシップ関数が満たされるべき要件は以下である。

1. メンバーシップ関数の底辺の幅が 0 以上である
2. メンバーシップ関数の頂点が一致しない
3. 隣り合う関数が重なっている、または接している
4. 一つの関数が他の関数を超越しない

以下は制約条件の記述例である。

FusionFunc 関数では、二次元列から、関数の頂点を取り出し、他の関数の頂点と一致しないかを bool 型で返す関数である。

CompareWidth 関数では、隣り合う関数が、重なっている、または接していれば true を返す関数である。

このような制約条件を他にも作成し、一つの操作にまとめ、テストをすることで、メンバーシップ関数の妥当性が確認できる。

```
functions
public FusionFunc:
  seq of (seq of real) -> bool
FusionFunc(list) ==
  forall i, j in set inds list & i < j =>
    list(i)(1) <> list(j)(1)
  pre forall i in set inds list
    & list(i)(2) > 0;

public CompareWidth:
  seq of (seq of real) -> bool
CompareWidth(list) ==
  forall i in set inds list & i < len list =>
    list(i)(1) + list(i)(2)/2 >=
      list(i+1)(1) - list(i+1)(2)/2
  pre forall i in set inds list
    & list(i)(2) > 0;
```

Listing 2 妥当性確認記述の一例

5.2 推論のながれ

グレード写像群により、入力値から各メンバーシップ関数のグレードの列が出力される。このグレード列を用いて曖昧性のない数値として出力するために、菅野の簡略推論法を用いて計算する [5]。簡略推論法は、演算速度が早いことと四則計算を使った式のため、解析が容易という利点がある。

以下に、入力値 x_1, x_2 が与えられた時の推論結果の出力 y を求める手順を示す。ここで b_i は、ファジィルールによって得られる実数値である。

手順 1 各規則の適合度 w_i を計算する。

$$w_i = \mu_{A_{i1}}(x_1) * \mu_{A_{i2}}(x_2)$$

手順 2 推論結果の出力 y を次式で計算する。

$$y = \frac{\sum_{i=1}^n w_i * b_i}{\sum_{i=1}^n w_i}$$

6 まとめと今後の課題

VDM++ 上で、ファジィ理論を用いた曖昧性判断を行うことにより、自然言語による仕様の曖昧さをより排除でき、システムの高品質化が期待される。

今後の課題として、ファジィ推論を VDM++ 記述に落とし込み、実際に曖昧な値を数値化し、様々な場面で応用できるようにすることである。

参考文献

- [1] ジョン・フィッツジェラルド, 他. VDM++ によるオブジェクト指向システムの高品質設計と検証. 翔泳社, 2011.
- [2] 牧田蒼斗, 和崎克己. 仕様記述言語 VDM++ を用いた制限付きオブジェクト予約システムの記述. 第 21 回情報科学技術フォーラム講演論文集, Vol. 1, No. A-002, pp. 95–96, 2022.
- [3] 菅原道夫. ファジィ理論とその応用. 化学と生物, Vol. 26, No. 9, pp. 562–567, 1988.
- [4] 伊藤修, 細川浩一郎, 栗原裕幸. 浄水プロセスのファジィ制御. EICA, Vol. 7, No. 3, pp. 23–29, 2002.
- [5] 日本ファジィ学会. ファジィ制御. 日本工業新聞社, 1993.