

代数的仕様の実用プログラムへの  
適用に関する考察

2R-7

大場克彦 金戸孝夫

(株)島津製作所 技術情報システム部

## 1 はじめに

コンピュータが社会の幅広い領域に浸透するにつれ、信頼性の高いプログラムを作成する必要性が高まっている。この必要性を満足する可能性のあるプログラム開発法の一つとして、代数的仕様記述法が注目され始めている。代数的仕様記述法は、数学的に厳密な定義が与えられた形式性を持ち、これを基に、機械的な仕様の検証やソースコードの生成が可能になる。また、等式論理を基礎にしているため、検証系や実現系を作りやすいという長所もある。しかし、現場レベルで使用するには、仕様を記述する立場からみると、

(1) 基礎となる理論を理解しないと仕様が書けないが、これが難しい。

(2) 理論に従って厳密に仕様を記述すること自体が、難解な作業である。

(3) 仕様が冗長になり、記述に時間がかかる、仕様に誤りが入りやすい。などの問題がある。一方、仕様を見る立場からは

(4) 仕様を理解しにくい。

という問題がある。しかし、代数的仕様記述法は、信頼性の高いプログラムを開発する上で、原理的に優れた方法なので、上記の問題を解決して現場レベルで利用できるようになることが望まれる。

## 2 現状

代数的仕様記述法を現場レベルで使用するためには、1で述べた問題点の

内、数学的な基礎理論を知らなくても、仕様を作成できることが、まず必要である。これを可能にしたものに、HOS<sup>(1)(2)</sup>がある。HOSでは、プログラム及びプログラムを構成する操作を関数で表現する。関数の等価関係は、1つの関数を2つの関数の合成関数として表現している。この関数の等価関係を、図式的に2進木として表している。2進木の分割は、関数があらかじめライブラリに登録されている抽象データ型の関数に到達するまで行われる。こうして作成された仕様は、検証系で検証され、ソースコード生成系でFORTRANなどの高級言語のソースコードに変換される。以上にみたようにHOSは、

① 基礎理論を知らなくても代数的な仕様記述が書ける。

② 仕様を階層的な図式で表現しているため、構造が分かりやすい。

③ 既存の抽象データ型をライブラリ化し、再利用可能にしている。

④ コンピュータ支援系を持っている。

などの特徴を有し、現場での実用レベルに一步近づいたものといえる。しかし、高級言語の制御構造が書けない、式を書くとき検証が行えないなど仕様記述能力が低いので、記述された仕様が冗長になる。その結果、仕様記述に時間がかかる(高級言語によるプログラミングの数倍)、仕様に誤りが入りやすい、仕様を理解しにくいなどの問題がある。

3 上記の問題の解決法に関する考察  
以上の結果を基に、代数的仕様記述法を現場レベルで使用する上での問題の解決方法について考察する。

(1) 理論を知らずに仕様が書ける

仕様の記述方法に関するマニュアルを作成し、このマニュアルに従えば理論を知らなくても、仕様が記述できるようにする。

(2) 書きやすい仕様記述法

仕様をすべて等式集合として記述しなければならないことが、仕様記述を難しくしている。この問題に対しては

- ① 自然言語で書いた仕様を等式集合に変換する変換系を用意する。
- ② 等式関係を関数で表現すると共に、等式集合を、相互の関連を考慮しながら、階層的な関式として表現する。という二つの解決方法が考えられる。

①の方法は、仕様記述の対象となるプログラムが多岐にわたるため、このような変換系の実現は容易でない。それに比べ②の方法は

- ・従来の階層的設計法に類似しているため、プログラマが馴染みやすい。
- ・関数名を自然言語で表現することにより、意味が理解しやすくなる。
- ・仕様の構成を構造的に表現できるので、理解性が高まる。
- ・コンピュータ支援を考えた場合、その実現が容易である。

などの利点がある。

(3) 仕様の冗長さを防ぐ

仕様の記述が冗長になるのは、次の3点が原因である。

- ① その仕様の定義に使用する、下位レベルの全ての抽象データ型の定義を行っている。これは、プログラムの中で使用するサブルーチンや命令の定義を、そのプログラムの中で行うのと同じである。
- ② 仕様記述能力が低い。
- ③ 代数的仕様記述法で記述される仕様

書は、従来のプログラム仕様書ではなく、テスト仕様書に相当するものと考えの方が妥当である。

上の原因に対して、次のような対策が考えられる。

- ① 検証系及び実現系をコンピュータ上に構築し、既に定義されている抽象データ型をデータベース化し、再利用可能にする。
- ③ 高級言語の制御構造や式を表現できるように、仕様記述能力を高める。
- ② 代数的仕様記述法で記述された仕様を、テスト仕様の設計に利用する。この結果、漏れや重複のないテスト仕様を得られる。

(4) 理解しやすい仕様の書き方

代数的仕様記述法で書かれた仕様を分かり難くしているのは

- ① 記述が冗長である。
- ② 仕様の記述方法の構造化が不十分。
- ③ 等式論理だけでは記述能力が不足。
- ④ プログラムの等式論理への不慣れ。などによる。これらは、既に述べた対策によって解決可能である。

4 おわりに

以上の考察結果を基に、代数的仕様記述法をベースにした、形式的仕様記述言語及びそのコンピュータ支援システムの開発を進めている。

文献

- 1) James Martin: AN INFORMATION SYSTEMS MANIFESTO, Prentice-Hall, INC.(1984) (成田光彰 訳: 管理職のための情報戦略、日経BP社(1986))
- 2) MARGARET HAMILTON, SAYDEAN ZELDIN: Higher Order Software - A Methodology for Defining Software, IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL.SE-2, NO1 (1976)