

# 抽象データ型に基づくプログラム設計 [ 1 ]

3 G - 8

大場克彦、金戸孝夫  
(株)島津製作所

## 1 はじめに

信頼性の高いプログラムを開発する手法の1つとして、代数的仕様記述法が注目され始めている。しかし、逐次処理型の実用レベルのプログラムの仕様を、平均的プログラマーが代数的仕様記述法を用いて書くのは難しい。平均的プログラマーが利用できるようになるためには、難解な基礎理論を意識しなくても、定められた手順に従って記述すれば、厳密であいまいさのない形式的な仕様を得られることが必要である<sup>[1]</sup>。これを可能にするために、抽象データ型を用いたプログラム設計法について検討した。

## 2 抽象データ型を用いた設計法

### 2.1 抽象データ型と操作

抽象データ型の基礎となっている概念は、①データと操作をパッケージ化し、データには定められた操作を通してしかアクセス出来ないというデータ型、②データ型から新しいデータ型を構成するデータ型の構造化、③仕様と実現方式の分離の3つである。抽象データ型の概念を使って、あいまいさのない厳密に定義された形式性を持つ仕様を記述する方法について考察する。

形式性を与えるためにプログラムおよび操作を、多変数入力多変数出力の関数形式で次のように表現する。

プログラム名 (入力1、…、入力n)  
→ (出力1、…、出力m) ①

操作名 (入力1、…、入力k)  
→ (出力1、…、出力l) ②

ここでプログラム名、操作名、入力変数、出力変数には自然な日本語表現を許すものとする。次に操作を、仮操作と終端操作に分類する。仮操作は、実現方式が定義されていない操作であり、仕様と実現方式の分離という概念に対応する。終端操作は、実現方式が定義されている操作である。さらに、終端操作を基本操作と定義済み操作に分ける。基本操作は、高級言語で用意されている基本データ型に対する操作であり、高級言語の文がこれに対応する。定義済み操作は、基本データ型や既に定義されているデータ型を組み合わせて新しく作ったデータ型に対する操作で、基本操作および自分より前に定義されている定義済み操作を用いて新たに定義した操作である。定義済み操作はデータ型の構造化という概念に対応する。終端操作がデータ型という概念に対応する。

### 2.2 プログラムの設計手順

上に述べた操作を使って、次の手順でプログラムの設計を行う。

(1) プログラムの名前と入力データ、出力データを①の書式にしたがって書き表す。プログラムの名前は、その機能が分かるようにつける。

(2) プログラムの機能を実現するために、どのような操作を施せばよいかを手順として表す。ここでは、操作名だけを順番に書き、その実現手段については考えない。すなわち、仮操作として操作を表す。

(3) 各仮操作への入出力を決め、②の書式で表す。

(4) 各仮操作が実現されれば、プログラムの機能が実現されることを確かめる。

(5) 仮操作のうち、終端操作で置き換え可能なものは置き換える。

(6) 全ての仮操作が、終端操作だけで表現される手順に変換されるまで以下の手順を繰り返す。

①仮操作を、新しい仮操作を使って手順として表現する。

②新しい仮操作が実現されれば、元の仮操作が実現されることを確かめる。

③新しい仮操作のうち、終端操作で置き換え可能なものは置き換える。

(7) データの名前と型を対応付けてデータ定義を行う。

以上の手順で得られたものを設計仕様と呼ぶ。

### 2.3 設計仕様の等式表現

いまプログラムをP、仮操作をK<sub>i</sub>、終端操作をS<sub>i</sub>で表す。2.2の手順で得られる設計仕様は、次のような等式表現で表される。

まず、(1) - (4)の手順により、プログラムと仮操作の関係を表す等式表現が得られる。

$$P \equiv K_1 K_2 K_3 \cdots K_n \quad \textcircled{3}$$

ここで、K<sub>1</sub>K<sub>2</sub>K<sub>3</sub>⋯K<sub>n</sub>は、仮操作K<sub>1</sub>、K<sub>2</sub>、K<sub>3</sub>、⋯、K<sub>n</sub>を、この順に施すことを意味する。③式は、プログラムの動作が操作K<sub>1</sub>、K<sub>2</sub>、K<sub>3</sub>、⋯、K<sub>n</sub>をこの順番に施したときの動作と同じであることを表す。ここで、同じ動作をする操作あるいは操作の列は合同であると定義し、記号" $\equiv$ "で表す。

(5)の操作により、③式の中の仮操作と終端操作の合同関係を表す等式集合が得られる。例えば、K<sub>1</sub>がS<sub>2</sub>、K<sub>3</sub>がS<sub>5</sub>と合同であるとき、次の等式関係が得ら

Program Design based on Abstract Data Type [1]

Katsuhiko OHBA, Takao KANETO

SHIMADZU CORPORATION

れる。

$$K1 \equiv S1 \quad \textcircled{4}$$

$$K3 \equiv S5 \quad \textcircled{5}$$

③式の中の終端操作に置き換えられない仮操作のそれぞれについて、新しい仮操作を用いて合同関係を定義する。例えば、K2を次のように定義する。

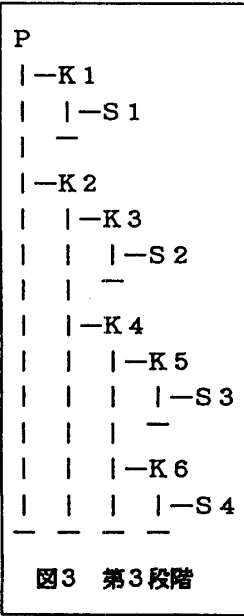
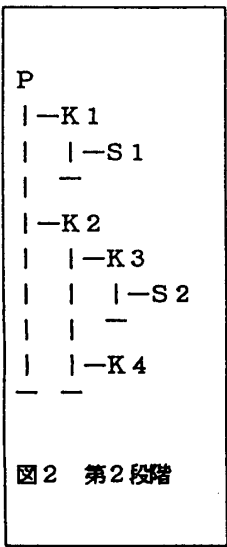
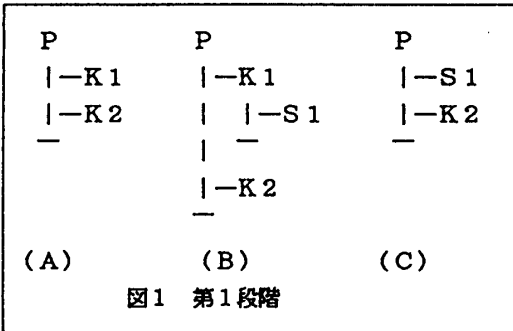
$$K2 \equiv K2\_1 K2\_2 \quad \textcircled{6}$$

ここで、K2\_1、K2\_2は、新しい仮操作である。いま、K2\_1がS2、K2\_2がS3と合同関係にあれば、

$$K2\_1 \equiv S2 \quad \textcircled{7}$$

$$K2\_2 \equiv S3 \quad \textcircled{8}$$

と表される。以下同様の手順で、仮操作を新しい仮操作を使って定義して行くと、仕様を表す等式集合が得られる。この等式集合は、左辺の項を右辺の項に書き換える項書換え規則と見なすことが出来る。設計仕様が意味を持つためには、この項書換え規則を適用したとき、設計仕様の中のすべての仮操作が終端操作の列に置き換えられることが必要である。



2.4 設計仕様の図式表現

2.2で定めた設計手順を用いて、代数的仕様(仕様を表す等式集合)を得る方法を示した。このようにして得られる等式集合は、冗長であり、また、必ずしも分か

りやすいものではない。ここでは、等式集合として表された設計仕様を、それと等価な木構造図で表す方法を示す。

まず、設計仕様が次のような等式集合とする。

$$P \equiv K1 K2 \quad \textcircled{9}$$

$$K1 \equiv S1 \quad \textcircled{10}$$

$$K2 \equiv K3 K4 \quad \textcircled{11}$$

$$K3 \equiv S2 \quad \textcircled{12}$$

$$K4 \equiv K5 K6 \quad \textcircled{13}$$

$$K5 \equiv S3 \quad \textcircled{14}$$

$$K6 \equiv S4 \quad \textcircled{15}$$

まず、合同関係⑨を図1(A)の様に表す。これは、操作K1、K2をこの順に施したものが、プログラムPと等価であることを表している。これに⑩の合同関係(仮操作K1が終端操作S1と等価である)を追加すると図1(B)のようになる。ここまでで、2.1の設計手順(1)-(5)を適用したことになり、設計の第1段階が終わる。設計の第2段階以降は、(6)の手順を適用することになる。仮操作K2を新しい仮操作K3、K4で表し、さらに、K3を終端操作S2と等価であると定義する。これが設計の第2段階で、合同関係⑪⑫で表される。設計の第2段階が終わると、図2の木構造図が得られる。設計の第3段階は、K4の詳細化である。これにより、合同関係⑬⑭⑮が追加され、木構造表現にすると図3となる。この例では、これで設計が終了する。もし、詳細化されていない仮操作があれば、すべての仮操作が詳細化されるまで、同様の手順で詳細化を続ける。

仕様の冗長さをなくすために、図1(B)の表現を図1(C)の様に表すことも可能である。

3 まとめ

抽象データ型の考えに基づき、仮操作と終端操作という概念を導入し、まず、プログラムを仮操作で表し、次に仮操作が終端操作で置き換えられるまで、仮操作の詳細化を続けるという設計手順により、代数的仕様(等式集合)を得る方法を示した。次に、同じ設計手順により、代数的仕様と等価な木構造図が得られることを示した。この木構造図は、従来のトップダウン設計で木構造図を作成するのと同じ手順で作成できるので、プログラマにとってなじみやすい手法である。しかも従来のトップダウン設計で得られる木構造図と違って、厳密に意味が定義された形式性を持っている。このことは、数学的な基礎理論を意識しなくても、厳密に意味が定義された形式性を持つ仕様を書くことが可能性であることを示している。さらに、反復や選択などの制御構造を含む仕様を記述できるように記述法を詳細化することにより、この可能性を現実のものにできると考えている。

参考文献

[1] 大場、金戸：代数的仕様の実用プログラムへの適用に関する考察、情報処理学会第40回全国大会(1990)