プログラム構造の抽出自動化の試み*

4Q−01

大段雅典†　武内亮　佐藤匡正‡
島根大学大学院総合理工学研究科§　島根大学総合理工学部¶

1. はじめに

一つの問題に対して作成された複数の同値プログラムをプログラム変異体と呼ぶ．こうした変異体を分析することにより，それを生じている背景にある要因や同等性の把握，またプログラムの評価などを行うことができる．プログラムが機構と構造の二つの部分から構成されるとする考えに基づき[1]，個々の変異体プログラムからその機構部分を削除し，構造を抽出し形式化することにより上の分析を行う．本研究における構造とはプログラムの本質の枠組みを与える部分をいい，機構とはプログラム中の繰り返し部分の回数制御や，選択の条件判定を行うプログラムの動作に一意性を与える部分をいう．

プログラム構造に類似した考えとして，サイクロマティック数[2]の提案がある．この提案では，プログラム構造をダイアグラムで表し，独立パスの数を求めてプログラムの複雑さを示している．本研究では，プログラムの複雑さや論理上の正当性，同等性を示すのではなく，プログラム仕様の最大限の枠組み関係を示すことによって，仕様との追従性を踏まえた人間の観察を支援することを目的としている．

本論文では，このプログラム機構の確定法による構造抽出自動化の方法を示し，この方法の応用として形式化した構造を用いてプログラム変異体の分析を行った結果について述べる．

* An Approach of Automatic Program Structure Analyzer
† Ohdan Masanori
‡ Takeuchi Ryo, Satou Tadamasa
§ Interdisciplinary Graduate School of Science and Engineering, Shimane University
¶ Interdisciplinary Faculty of Science and Engineering, Shimane University

2. プログラム構造の抽出法

2.1. 基本的な考え方

プログラム構造を抽出するために，機構部分を確定し削除することによって間接的に抽出する方式を考える．プログラムにおいて，機構はごく限定された部分を占めるから，構造を直接抽出するよりも機構を確定し削除したほうが容易である．

機構の基本要素は，命題文において一意性を与えている文である．したがって，命題を全て削除することにより構造の候補を確定することができる．ところが，この方法のみでは適切とは言えない．例えば，フラグやカウンタなど命題文に意味付けを与えるためだけの文が，プログラムの本質部分として存在するため，プログラムとしての意味上に不都合を生ずることになる．構造を得るには，さらにこれらを取り除かなければならない．しかし，機構と構造の両方を形作る要素処理も存在するため，全ての命題文を取り除くと，機構と同時に構造を形作っている文も除外される惧れがある．

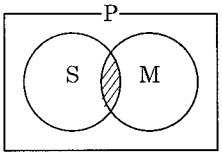

図1　構造(S) と機構(M)

プログラム文の集合をP, 構造を形作っている文をS, 機構を形作っている文をMとして，この様子を図1に示す．

いま，Sを抽出するには，Mを識別し削除する際にMに含まれているSの文が対象から除外されればよい．

$$S = P - (M - (S \cap M))$$

2.2. プログラム構造抽出アルゴリズム

構造を抽出するために，上のM部分およびS∩M部分を確定し，Sを求める．

1) M部分の確定

命題記述に用いられている変数名，関数名を得る．これら変数名，関数名をもつ要素処理文によって機構Mが構成されている．したがって，命題文とその関連

処理によってMは確定できる．

2) S∩M部分の確定

上で確定したMを構成している変数（機構変数）のうち，Sをも構成している変数（構造変数）を確定する．その変数を含む処理がS∩M部分となる．そのために，変数の設定・参照順序関係に着目し，以下の手順で判定された変数をSの要素とする．

① 設定・参照順序関係の明確化

次項で述べるように変数の設定・参照順序関係の解析を行い，その結果を得る．

② 構造変数の判定

①に対して，以下の構造変数の条件と照らし合わせて判定する．

構造変数の条件：

命題文以降で，命題記述以外での参照がなされている変数．

3) Sの抽出

MからS∩Mを除外した処理を，プログラム文から削除することで得られる．

2.3. 設定・参照順序関係の解析法

以下に設定・参照とその順序関係を解析する手順を示す．

① 命題記述文に用いられている変数名，関数名を列挙する．

② それらに対して表1に示す細分化された設定・参照操作の状況を，プログラムの処理の流れに沿って表す．

③ ②への追加情報として，他の処理との値の受け渡しを明らかにする．

表1 設定・参照の定義

設定	S	参照	R
関数引数	H	命題参照	P
関数呼び出し	O	カウント処理	C

2.4. 実現方法

ここで，このアルゴリズムを処理系として実現するための制約条件を挙げる．

- 構造のない変数

プログラムを表面的にとらえるため，内部に複数の値を格納しているレコード型や構造体と呼ばれる変数や，値が格納されている番地を持つポインタ変数などは対象外とする．

- 内部変数

単体内の変数に限定する．

- 関数参照

関数などの戻り値が直接的に命題記述文で参照されている場合は，作業用変数を用いて関数値を退避させる．これは，関数が機構と判断されるのを避ける措置である．関数以外で複数の処理が命題記述文で行われている場合も同様とする．

3. プログラム変異体の分析

変異体プログラムP1，P2に構造抽出法を適用し，形式化[2]を行ったものが以下の数式となる．これは，選択や繰り返しの一意性をなくしたそれぞれの処理の連なり方を表している．

$$P1 = A \cdot (H \cdot E \cdot G + H \cdot F \cdot G)^*$$

$$P2 = A \cdot (J \cdot F \cdot H + J \cdot G \cdot H)^*$$

各プログラムの処理の対応を以下の表に示す．

表2 処理対応表

P1	P2	処理内容
H	J	x=i%2
E	F	iを表示
G	H	i++

この表から，数式P1=P2であることがわかる．したがって，P1とP2のプログラム構造は同等であることが言える．

4. おわりに

機構の主要素である命題に用いられている変数に着目して構造を抽出する方法を案出し，その処理系の実現を行った．

参考文献

1) Satou Tadamasa: “Universal Form for Program Iteration Structures”, pp.240-243, ICCIT 2000

2) McCabe, T.J.: A complexity measure, IEEE Trans on Software Engineering, SE-2 (4), 1976

3) 佐藤匡正：「HCP図法で記述されたプログラム解法のS代数による定式化」 情報処理論文誌 Vol.27, No.6