繰返し本体コードの正しさ検証のための会話型支援機能の検討と試作

2000MT035 神野 義春　　2000MT095 冨田 光治
指導教員　真野 芳久

1　はじめに

プログラムの正しさを検証するのに種々の方法がある。プログラムの繰返し部分の正しさ検証については、ループ不変条件という繰返しにおいて常に成立する条件を用いた証明法がある。ループ不変条件は論理記号を用いて詳細かつ正確な表現が求められ、利用者にとって煩雑であり、大きな負担になる。そこで、ループ不変条件を図式表現したループ不変図式を用いる方法を考える。ループ不変図式を用いることで視覚的にループ不変条件を理解でき、繰返しの正しさを検証する上での手助けとなる。ここでは、プログラムコードとループ不変条件を用いて、図式化されたループ不変条件を利用した繰返しコードの正しさ検証のための会話型支援機能の実現にむけて議論する。

2　プログラムの正しさの検証に向けて

プログラム特に繰返し部分について正しいことを示すには部分正当性と停止性について証明する必要がある。ここでは部分的正当性についてのみ述べ、その意味で‘正しさ’という用語を用いる。

2.1　ループ不変条件

ループ不変条件は繰返しの各時点で真となっているという性質を持ち、プログラムの正しさを検証する上での重要な概念である。繰返しを表す流れ図におけるループ不変条件は図１中のＩで示される。

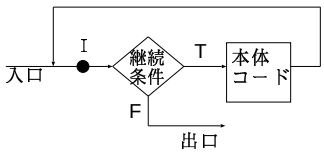

図１　繰返しの流れ図とループ不変条件

2.2　繰返し１回の差

本研究では繰返しコードの正しさを確かめるために、ループ不変条件が繰返し内で常に成立するという性質を利用した正しさの検証を行う。

これは、繰返し中のある一般的な状態と、そこから繰返しを１回進めた状態でループ不変条件が成り立っていることを調べることで正しさを判断する方法である。

ここで繰返しを１回進めるとは、繰返しのある途中の地点から繰返しが進んでいき、次にまた同じ地点に来るまでのことである。

ある一般的な状態とそこから繰返しを１回進めた状態とで、ループ不変条件が成立していることを確かめられれば、繰返しが２回・３回…と進んでいっても同じ状況であると考えられ、ループ不変条件が繰返し中常に成り立っている条件であることが確認できる。

繰返し中のある一般的な状態と、繰返しを１回進めた状態を進行図式と呼び、それらの違いを繰返し１回の差と呼ぶ。この差を調べることが繰返しの正しさを確かめる上で重要であり、これを支援するシステムの作成を行う。

3　ループ不変図式を利用した正しさの検証

3.1　ループ不変図式の利点

ループ不変条件は論理記号を用いて詳細かつ正確な表現が求められる。これは利用者にとって煩雑であり、論理式などの扱いに熟達していない者にとって理解しづらい。そこでループ不変条件を図式表現したループ不変図式を用いる方法を考える。ループ不変図式とは、ループ不変条件のデータ構造を表す図とそこで成り立っている要素の関係を示す式を用いる表現形式である。ループ不変図式を導入し図式化されたループ不変条件を視覚的にとらえることで、ループ不変条件という概念への理解性を高めることができると考えられる。

3.2　ループ不変図式による検証法

ループ不変図式を用いた繰返し部分の正当性の検証は次のように支援される。

1. 事前条件より、繰返し直前の状態のループ不変図式が成り立っていることを確かめる。
2. ループ不変条件が成り立っている地点のループ不変図式と、繰返しを１回進めた状態のループ不変図式とを見比べることにより、ループ不変条件が繰返し中で常に成り立っている条件であることを確かめる。
3. 繰返し終了直後の状態のループ不変図式より事後条件が成り立っていることを確かめる。

繰返しの直前でループ不変条件が真であることと、繰返しの終了時に事後条件を満たしていることを、論理式を使用しないで確かめることができる。さらにループ不変条件が繰返し中常に成り立っている条件であることの確認ができ、繰返しコードの正しさの検証ができる。図式による表現では、ループ不変条件の厳密な意味での表現はできないが、人にとって繰返しの正しさの直観的な理解ができ、より実践されやすいことが期待できる。

また図式による表現では、繰返しコードの誤りやループ不変条件の変更があった場合等の検討作業に用いるこ

とができるという利点もある。

3.3 ループ不変図式による検証例

ここでループ不変図式により繰返しの正しさを検証する例を、『大きさ N の配列 a[N] についてそれぞれの要素の和を求める』という問題について示す。

ある途中の状態のループ不変図式とそこから繰返しが 1 回進んだ進行図式とを縦に並べた図式は**図 2** になる。2 つの図式が本質的に同じ状態を表現していることから、ループ不変条件が実行によっても '不変' であることを確認できる。

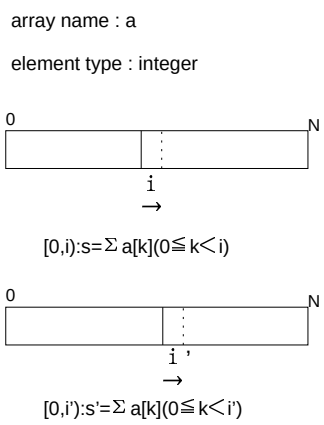

図 2 配列の和を求める問題のループ不変図式と進行図式

4 検証システムの実現にむけて

本システムの目的は繰返し部分の正しさ検証のためにループ不変図式を利用した支援を行うことである。ある途中の繰返しの状態と繰返しが 1 回進んだ状態との差を求め、それを図式にすることで繰返しの正しさ検証を支援する。繰返し 1 回の差の求め方とそれを用いたシステムの実現について検討する。

4.1 システムの設計

システムの設計方針を示す。

1. ループ不変条件をプログラム本体から機械的に導くのは困難である。ここでは、ループ不変条件は利用者が与えるものとし、作成するシステムはプログラムコードとループ不変条件の入力から図式化を行い、繰返し部分の正しさ検証を会話的に支援していく。
2. 本研究で扱うデータ構造は配列のみとする。
3. 扱うプログラミング言語は C 言語で、繰返し部分の検証という本研究の目的を損なわない程度に制限を設ける。

4.2 システムの構成

本システムは大きく以下のように分けることができる (**図 3**)。

- 利用者から与えられたプログラムコードを解析し、以降のフェーズで扱いやすいように構文木と記号表でプログラムコードのデータを表すようにする解析部
- 利用者から与えられたループ不変条件と解析されたデータを用いてある状態から繰返しを 1 回進めた状態を作り出す実行部
- 解析部と実行部で得た情報を用いて、実際に図式化を行う図式作成部
- 入出力などを行うインターフェース部

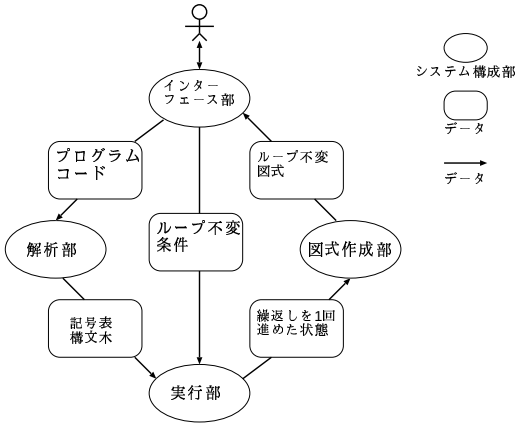

図 3 システムの構成

解析部を冨田、実行部を神野が担当する。他の部分については、上記重要部分の実験ができる程度の暫定版を協力して検討・作成する。

5 解析部

5.1 解析の流れ

プログラムを解析し図式化を行うために、プログラムの構造を表す構文木と記号表を作成する。解析の流れを**図 4** に示す。

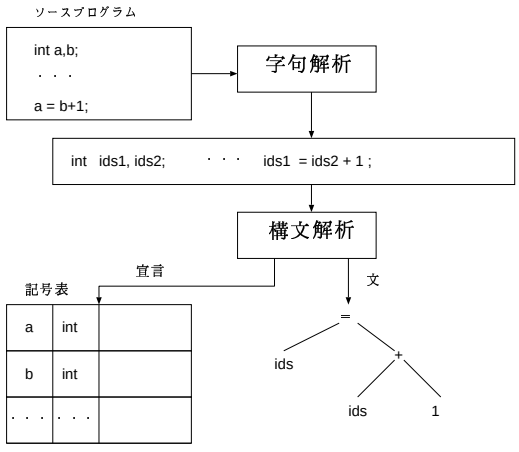

図 4 プログラム解析の流れ

5.2 記号表の構成

C では変数を用いる場合には、その名前を宣言しなければいけない。変数の名前やその他の情報を表の形で保持する。次のプログラムコードから作られる記号表を図 **5** に示す。

```
int g, P();
void F(int n)
{
  int a;
  …
```

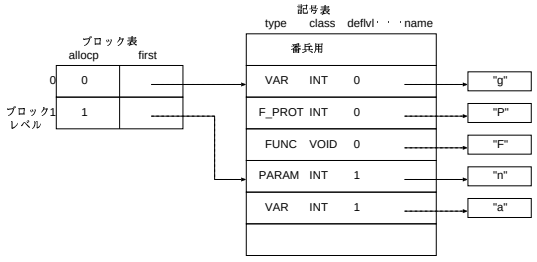

図 5 記号表とブロック表

記号表のエントリは次のフィールドからなる構造体として定義する。

- type: 識別子のデータ型
- class: 識別子の種類を表す値
- deflvl: 識別子が宣言されたときのブロックレベル
- Nparam: 関数名の場合に、引数の数を示す
- name: 識別子文字列へのポインタ
- dim: 配列の場合に、配列の次元数を示す
- dlist: 配列の場合に、配列の大きさを示す

5.3 構文木の構成

プログラム全体を 2 分木の構文木として表現する。2 分木の構成単位の節点は、次のフィールドからなる構造体として定義する。

- left : 左部分木へのポインタ
- right: 右部分木へのポインタ
- Op : 節点の種類
- type : 節点の型
- value: 節点が定数や識別子などの時にその値や識別子へのポインタを記憶しておくための共用体

6 実行部

6.1 繰返しを 1 回進めた状態

ループ不変条件中の情報は、具体的な値を用いた関係だけではなく、より一般的な関係式によっても表現される。そのため繰返し 1 回の差を具体的な値として求めることは一般的にできない。

ここでプログラムコードやループ不変条件は、変数や数値などの記号の並びであると考えられる。プログラムコードを解析し、代入文などで記号が変化しており、それぞれの繰返し中の文により変化する記号を求めることで、記号により繰返しが 1 回進んだ状態を表すことができる。

記号により表現したものは具体的な値はもたないが、大小関係は記号の性質によりわかることがある。さらに記号による表現でもループ不変条件をループ不変図式で表現するときに必要な情報 [2] である要素間の関係などが満たされているかどうかを判断することが十分にできる。この方法で繰返しを 1 回進めた状態をループ不変図式にしたときにループ不変条件が成り立っているかを確認できる。このように繰返し中のそれぞれの文で変化する記号を求め、繰返しを 1 回進めた状態を求めることで繰返しの正しさ検証ができると考える。

6.2 実行部の処理

実行部では解析部で得られた構文木を用いて解釈実行し、繰返しを 1 回進めたときに変化する変数の値を求め、図式作成部に与える。

本研究で目的としている繰返しの正しさ検証をループ不変図式により行う方法では、繰返し途中と繰返しが 1 回進んだ状態が同じ状況であることが視覚的に確認できることで正しさ検証を支援できると考える。

構文木と記号表を使い、繰返し 1 回のそれぞれの記号の変化を求める。構文木の節点の種類ごとに処理が異なる。変数の値が変化する場合の処理は次の 3 つがある。

1. 代入：変数の値を代入する記号列でおきかえる。可能な限り簡略化された記号列で表す。
2. 選択：場合分けの処理ごとの記号列の変化があるので複数の記号表を作る。新たにできた記号表の数の図式を作成することになる。
3. 繰返し：繰返しにより変化する変数の値を、繰返し終了時の値を表す仮の記号で置き換える。その繰返しの変化と全体の繰返しとの変化から繰返しを 1 回進めたときの変数の値を表す。

実行部での処理は、解析部で得られた構文木の while 文以下をみていき、それぞれの文により変化する記号を節点の種類別の処理で求める。ループ不変条件に必要なそれぞれの情報を入力し、それぞれの情報の繰返し 1 回の差が求まり、それらを図式作成部へと渡す。

7 解析部と実行部の実験

ループ不変図式を使った具体例を、『各要素に赤 (Red)、白 (White)、 青 (Blue) が入っている配列 $A[0..N-1]$ を、Red、White、Blue の順に並び変える』オランダ国旗問題について考える。

プログラム (**図 6**) を解析し、解析部は while 部分に対する構文木 (**図 7**) と記号表 (**表 1**) の 1 つを作る。

ここで、この問題のループ不変条件は、「$1 \leqq r \leqq w \leqq b+1 \leqq N$ で $A[0]$〜$A[r-1]$ に Red が、$A[r]$〜$A[w-1]$ に White が、$A[b+1]$〜$A[N-1]$ に Blue がある」である。

```
int r, w, b;
r = w = 0; b = N - 1;
while(w <= b){
  if (A[w] == Red  ) {
    swap(&A[r], &A[w]); r++; w++; }
  else if (A[w] == White)
    w++;
  else   /*A[w] == Blue*/ {
    swap(&A[w], &A[b]); b--; }
}
```

図 6　オランダ国旗問題を解くプログラム

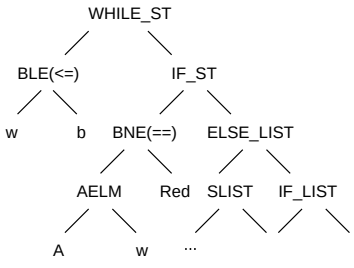

図 7　図 6 のプログラムの構文木

ある状態から繰返しを 1 回進めた状態を考えてみる。この問題では要素の色により 3 通りの状態が考えられるため、3 通りのループ不変図式が存在する。このことは構文木をみることで if 文による 3 通りの場合分けがあることからわかる。Red のとき、繰返しを 1 回進めた状態の変数の値を**表 1** に示す。**表 2** 中の繰返しを 1 回進めた変数の値は、変化を記号列により表している。r' では、繰返しを 1 回進めた r の値 r' は $r+1$ という記号列で表す。配列 A' の変化は、変化する配列の要素について記す。さらに構文木と記号表を実行部を通して、Red のとき r と w の値が 1 増加し、White のとき w の値が 1 増加、Blue のとき b の値が 1 減少することがわかる。それぞれの添字の関係 $r \leqq w \leqq b+1$ と添字で区切られる範囲の性質 $A[0]$〜$A[r-1]$ に Red、$A[r]$〜$A[w-1]$ に White、$A[b+1]$〜$A[N-1]$ に Blue からそれぞれの状態の進行図式が**図 8** となる。

このように図式化することにより、どの 3 通りの状態も $1 \leqq r' \leqq w' \leqq b'+1 \leqq N$ において $A[0]$〜$A[r'-1]$ に Red が、$A[r']$〜$A[w'-1]$ に White が、$A[b'+1]$〜$A[N-1]$ に Blue があることが**図 8** を見ることから理解できる。すなわち繰返しを 1 回進めてもループ不変条件が成り立っていることが確かめられ、繰返しコードの検証が行えた。

表 1　図 6 のプログラムの記号表

name	class	dim	dlist	···
A	I_ARY	1	10	
r	INT			
w	INT			
b	INT			

表 2　オランダ国旗問題の変数の値

名前	繰返しを 1 回進めた変数の値
A'	$A[r] : A[w]$　$A[w] : A[r]$
r'	$r+1$
w'	$w+1$
b'	

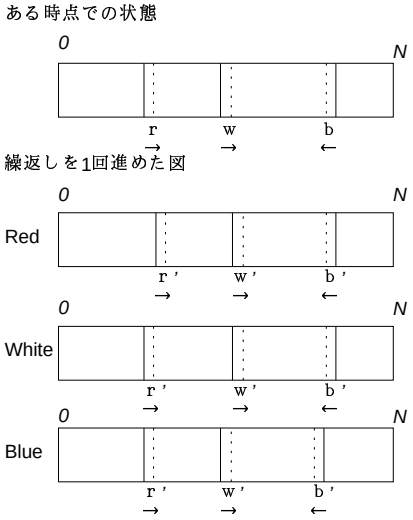

図 8　ループ不変図式と進行図式

8　おわりに

ループ不変条件を用いることによる繰返し部分の部分正当性の確認方法について述べた。また、繰返しの停止性についても図式によって示すことができるならば、より一層の正しさ検証ができると考えられる。さらに、配列以外のデータ構造により表されるループ不変条件をも扱えるようにすることでより一般的な検証システムになると考える。

参考文献

[1] David Gries：The Science of Programming, Springer-Verlag, 1981. 筧 (訳)：プログラミングの科学, 培風館 (1991).

[2] 古田壮宏, 真野芳久：説明文書としてのループ不変条件を利用した文芸的プログラミング, 南山経営研究 第 17 巻 第 1・2 号 (2002).

[3] 原田賢一：コンパイラ構成法, 共立出版 (1999).