1Z-1

プログラミング初学者のための学習支援システムにおける正誤判定方法の提案

井田　泰平　　小野　智義　　菊地　優太　　荒井　正之

帝京大学理工学部情報科学科

1. はじめに

我々は，プログラミング初学者の学習過程を次のように考えている．(1)プログラム全体構造，入出力文，変数などの理解．(2)連接，条件判断処理，反復処理などのアルゴリズムの基本構成要素の理解とそれらを使ったプログラムを 1 行 1 行トレースする能力の習得．(3)アルゴリズムの基本構成要素を用いて，プログラムを作成する能力の習得．(4)各種アルゴリズムやデータ構造をプログラムとして記述する能力の習得．これまでに(2)を支援するシステムの開発を行った[1]．本稿では，(3)を支援するために，学習者が記述したプログラムの正誤判定を行う方法について提案する．

2. 「プログラミング初学者のためのトレース能力の習得を目的とした学習支援システム」の概要

本研究では，前述の「連接，条件判断処理，反復処理などのアルゴリズムの基本構成要素の理解とそれらを使ったプログラムを 1 行 1 行トレースする能力の習得のため」に作成したシステム[1]をベースに記述問題の正誤判定を行う方法を提案する．このシステムの特徴として，出題に使用されたプログラムまたは学生が作成したプログラムの可視化が可能なこと，フローチャートが自動生成されること，実行行とそれに対応するフローチャートが 1 行 1 行赤くなり，それにあわせて変数の値も変わること，などがある．

2.1 ユーザインタフェース例

図 1 にシステムによって可視化されたプログラムの一例を示す．図 1 の A にはソースプログラム，B には変数の値，C にはフローチャート，D には実行行の説明が表示される．

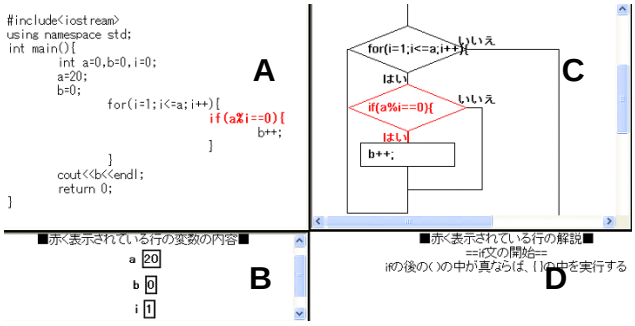

図 1　学習者用ユーザインタフェースの一例

2.2 プログラミング可視化の実現方法

ソースプログラムのトレースに必要な可視化情報の表示は，ソースプログラム，スケジュールデータ(SD)，フローチャートデータで実現している．図 2 に SD のデータの一部を示し，SD のデータ構造を説明する．

```
⋮
<no>4_i=1,n=4,ans=1
<for_s_1>5
<no>5_i=1,n=4,ans=1
<no>6_i=1,n=4,ans=1
<for_e_1>7
<for_s_2>5
⋮
```

図 2　スケジュールデータの一例

SD は実行パスと変数の値の情報が格納されているものである．例えば図 2 の<no>は実行行，<for_s><for_e>は for 文の開始と終了，その右側は変数の内容を示している．本研究では SD を用いて正誤判定をする方法を提案する．

3. 記述判定方法

3.1 処理手順

記述問題の正誤判定の流れを図 3 に示す．

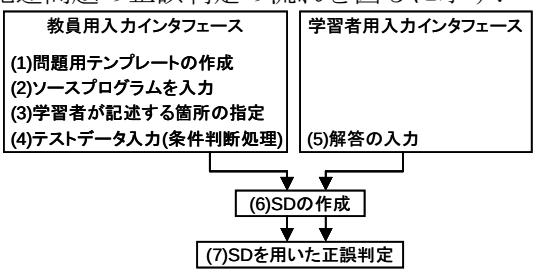

図 3　処理手順

処理(1)～(7)の実装方法について以下に詳しく説明する．

「a program evaluation method for the novice programmers' learning support system」
Taihei IDA, Tomoyosi ONO, Yuta KIKUCHI, Masayuki ARAI, School of Science and Engineering, Teikyo University

3.2　処理(1)～(5)の実装方法

教員は図 3 の処理(1)に示したようにまずは問題用のテンプレートを作成する．次に図 3 の処理(2)に示したように問題に用いるソースプログラムを入力する．ソースプログラムの入力画面の一例を図 4 に示す．

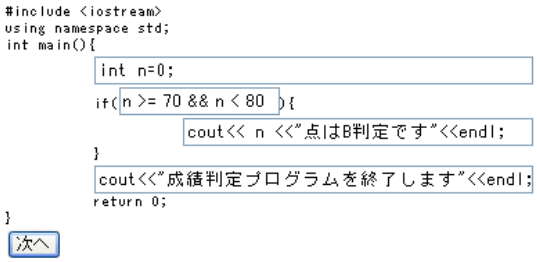

図 4　問題用テンプレートを用いたソースプログラム入力

次に教員は図 3 の処理(3)に示すように解答箇所の指定を行う．指定箇所が白抜きとなり，学習者はここに解答を記述することになる．一例を図 5 に示す．

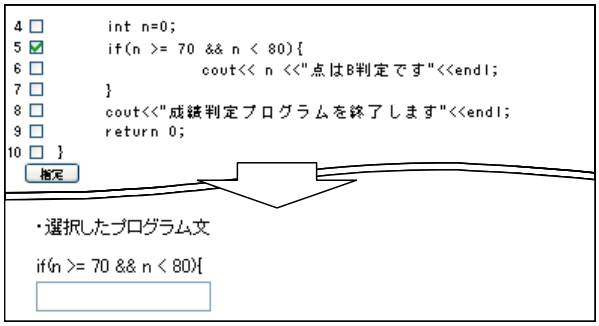

図 5　学習者が記述する箇所の指定

図 3 の処理(4)に示したように，問題に条件判断処理文が含まれる場合には，テストデータの入力を行う．条件判断において，正誤を判定するには，条件が真になる場合や偽になる場合など，種々のケースについて評価を行わなければならない．テストデータはこれらの評価に使用される．例えば，図 5 の問題のテストデータの場合，「n=69,70,71,79,80,81」のように入力する．

処理(3)で指定した箇所に対して，図 3 の処理(5)に示したように学習者は解答を記述する．

3.3　処理(6)「スケジュールデータの生成」の実装方法

図 3 の処理(6)に示した「スケジュールデータの生成」では，前の処理までに得られた教員および学習者のソースプログラム，テストデータからテストデータと同数の SD を生成する． テストデータが入力されない場合は，SD が 1 つだけ生成される．SD の生成手順を図 6 に示す．

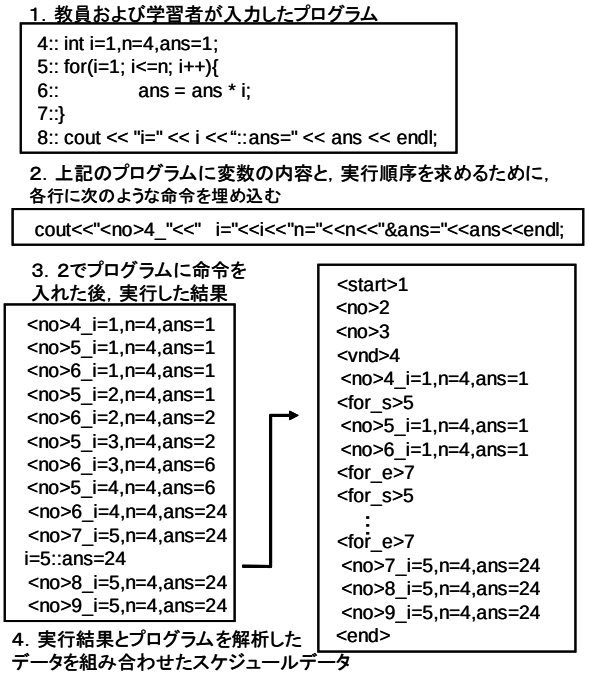

図 6　スケジュールデータ生成手順

3.4　処理(7)「スケジュールデータを用いた正誤判定」の実装方法

図 3 の処理(7)に示した「スケジュールデータを用いた正誤判定」について，図 7 を例に説明する．

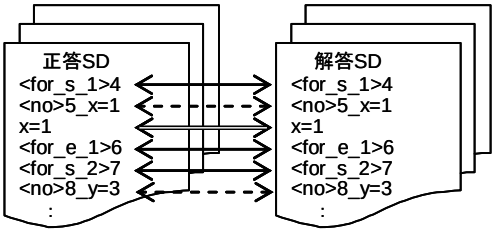

図 7　スケジュールデータの比較による正誤判定

教員と学習者が入力したそれぞれのソースプログラムに対して，テストデータと同数の SD を生成し，対応する SD を比較することにより正誤判定を行う．図 7 に示すように，対応する 2 つの SD において実行パス，変数の値，実行結果が同じである場合は正答，そうでない場合は誤答と判定する．図 7 において，実線は実行パスの比較，破線は変数の値の比較，二重線は実行結果の比較を意味する．

4.　おわりに

プログラムの記述問題の正誤判定を行うために，実行パスと変数の値が格納されたデータを用いる方法を提案した．今後の課題として，実験によるシステムの有効性の確認，機能の追加などが上げられる．

参考文献
[1]山崎倫巳，荒井正之：プログラミング初学者のトレース能力の修得を目的とした学習支援システムの開発，情報処理学会第 68 回全国大会， 4V-11，(2006)