

科学研究費助成事業 研究成果報告書

平成 28 年 6 月 3 日現在

機関番号：12601

研究種目：挑戦的萌芽研究

研究期間：2013～2015

課題番号：25540160

研究課題名(和文) 初学者のプログラミング教育におけるプログラミング能力の把握とその改善

研究課題名(英文) Research on Understanding of Programming Skills for Introductory Programming Education

研究代表者

関谷 貴之 (SEKIYA, Takayuki)

東京大学・情報基盤センター・助教

研究者番号：70323508

交付決定額(研究期間全体)：(直接経費) 2,500,000円

研究成果の概要(和文)：プログラミング初学者は、条件や繰返しなどの基本的なプログラミング要素を正しく理解していないことがある。そこで、初学者向けのプログラミング教育において、簡単なプログラムの実行結果を答えさせることで、変数の値の変化を追うトレーシング能力を問う小テストを開発した。また、このような小テストを実施するために、学習管理システムと連携して動作するオンラインツールを開発した。この小テストで見られる初学者の特徴的な誤りを「誤答パターン」と呼び、誤答パターンに当てはまる学生について、プログラミング全体の能力を問う定期試験との成績の相関を調査した。その結果、成績に影響のある誤答パターンが存在することが分かった。

研究成果の概要(英文)：Novice programmers' understanding of conditional and loop constructs are often incomplete. They seem to understand a single conditional or single loop, but fail to understand the combination of them. We propose a method for finding misconceptions underlying this failure. We first developed a tracing quiz set to locate the exact points at which students will fail. Second, we identified some misconceptions from experiments on five courses. Third, to use and validate these misconceptions, we developed an interactive test system which showed the correct answers to the students and requested them to describe their explanations. The experiments showed that some misconceptions affected the overall performance of the students.

研究分野：高等教育を支援する情報システムの開発と運用及びカリキュラムの分析

キーワード：プログラミング教育 誤答分析 トレーシング

1. 研究開始当初の背景

プログラミングの初学者に対する教育に関して、その能力に注目した様々な研究が行われている。例えば、Lopez らはプログラムを読むスキル(リーディング)、変数の値などの変化を追うスキル(トレーシング)、書くスキル(ライティング)に注目して、期末試験の学生の解答を統計的に分析することで、スキル間の階層関係やトレーシングとライティングとが強く関わる事を発見している(引用文献①)。我々はトレーシング試験と期末試験の結果の比較から、for ループを正しく解釈できない学生の成績が良くない傾向がある事を明らかにしている(引用文献②)。しかし、プログラムを書けるよう教育する事は未だ容易ならざる問題である。

本研究では、様々な設問に対する初学者の解答結果から誤答パターンを発見して、これを手続きとして表現した上で、当該パターンに従った誤答を予め生成する。それによって、学生の解答結果がどの程度の割合で誤答パターンに一致するか、また一定割合以上の設問で誤答パターンに一致する学生のプログラミング試験結果を比較分析する事で、誤答パターンとプログラミング能力との相関を得る。

2. 研究の目的

本研究は、基礎的な知識や能力を問う問題を用いて、プログラミング能力を予測する事を目的とする。そのためには、基礎的な問題で見られる初学者の誤り — これを我々は「誤答パターン」と呼ぶ — とプログラミング能力との関係を分析する必要がある。既に我々は for ループに関する知見を得ているが、他のプログラム構成要素とプログラミング能力との間で、同様の事例がないかを調査する。次に、発見した事例と同種の問題をオンラインテストとして実施して、学生のプログラミング能力を見極めるシステムを構築する。以上が研究期間内に実施する範囲である。

本研究の特色は基礎的な問題での誤答パターンに注目する点である。本研究が成功して誤答パターンを見付けるシステムを完成できれば、日々の授業についていけなくなる学生を早期に発見して、彼らが弱点を克服する手助けをし、より多くの学生がプログラミング能力を身に付けられる。プログラミング能力を持った人材の輩出にも繋がり、あらゆる産業でソフトウェアの重要さが増す現在では、社会的に大きな意義を持つ。

3. 研究の方法

(1) 我々は過去に予備実験として基本的な繰返し構文である for ループと条件分岐 if-then を様々な組み合わせたコードを用いた複数の設問からなるトレーシングの試験を実施している。例えば、図1の左のコードで a=3 及び a=4 とした時の出力として、

3 及び 4 とする学生が約 42% いた。これは for ループによる反復を解せず、図1の右のコードが実行されるかのような誤った解釈をしていると推測される。このように、同様の構造を持つコードで共通して見られる誤った解釈を本研究では**誤答パターン**と呼ぶ。

```
def a3(a)
  ans = 0
  for i in 1..a
    ans = ans + a
  end
  p ans
end

def a3(a)
  ans = 0
  ans = ans + a
  p ans
end
```

図1: a3.rb(左) とループを無視した誤った解釈(右)

このような誤答パターンを再現するには、初心者の思考過程と同様にコードを解釈する一種のインタプリタを構築する方法が考えられる。しかし、図1に示したような比較的短いコードであれば、想定する誤答パターンに基づいてコードを変換することでインタプリタの代りとなると考えた。そこで、本研究では同様の比較的短いコード10個について、それぞれ2種類の入力を与えた際の出力を問う設問20個からなる小テストをプログラミング初心者を受験してもらい、その解答結果と誤答パターンに対応する変換を適用して得られる誤答とを比較した。

誤答パターンとしては、表1に示す過去の予備実験の結果から推測した基本的な誤答パターン(以下「基本誤答パターン」という)及び、その基本誤答パターンを組み合わせた複合誤答パターンを生成して、それらの誤答パターンを適用して得られる誤答と小テストの受験者の回答とを比較している。

なお、誤答パターンの妥当性を検討するために、小テストの実施直後に各設問の正答を示し、誤った設問に対してその理由を説明させるアンケートを実施している。また、誤答パターンから推測されるプログラムのトレーシング能力と成績との関係を分析するために、それぞれの講義で実施した期末試験の結果との比較も行った。

(2) 本研究でプログラミングのトレーシング能力を測る小テストは、ウェブベースのオンラインテスト用のツールで実施している。本ツールは、トレーシング問題を受験者に解答させた後、正答を示すと共にその解答理由を問うアンケートを実施する仕組みになっている。

一方、教育活動を支援するシステムとしては学習管理システム(LMS: Learning Management System)が普及している。多くのLMSはオンラインのテストやアンケートの機能を有するが、本ツールのようにテストとアンケートを組み合わせる実施したり、プログラムの実行結果を解答させたりするには、機能が不足する。このような場合、既存のLMS

パターン名	説明	設問数
AETI	不等号のみの条件式 (「より大きい」或いは「より小さい」) に、誤って等号を補って解釈 (「以上」或いは「以下」) と解釈。	5
CSOI	不等号の向きを誤って解釈。	5
CVIC	for ループ内の if 文の条件式のなかに現れる <u>for 文の終端の変数を</u> 、 <u>for 文の制御変数</u> がそこにあると解釈。	3
CVIC2	CVIC とは反対に、for ループ内の if 文の条件式のなかに現れる <u>for 文の制御変数</u> を、 <u>for 文の終端の変数</u> がそこにあると解釈する。	3
CVIL	for ループ内に現れる <u>for 文の終端の変数</u> を、 <u>for 文の制御変数</u> と解釈する。	8
CVIL2	CVIL とは反対に、for ループ内に現れる <u>for 文の制御変数</u> を、 <u>for 文の終端の変数</u> と解釈する。	6
CVISL	CVIL と同様だが、for ループの中が代入式 1 個の場合のみ、 <u>for 文の終端の変数</u> を <u>for 文の制御変数</u> と解釈する。	4
NFL	for ループを解釈しない。	8
RAA	答えとなる変数を配列と解釈。	13

を改修することも考えられるが、LMS はオンラインテスト以外にも様々な機能を有しており、一部の改修によって全体に影響を及ぼすことで生じ得る不具合が懸念される。

そこで本研究では、LMS とその周辺のツールとの間で利用者のデータを交換するなど、異なるシステムを一体的に運用するための標準規格 IMS Learning Tools Interoperability (以下「LTI」という) に準拠した Tool Provider として動作するよう、本ツールを改修することとした。これによって、LTI に準拠した LMS であれば本ツールを用いることが、比較的容易になる。

4. 研究成果

(1) 図 1 に示した for ループと if-then で構成されるトレーシングに関する小テストを、本研究期間以前も含めて 5 年間実施してきた。

小テストを受験したのは、Ruby を用いた初心者向けのプログラミングの講義の受講者である。講義は、多くの大学で一般的な 90 分授業 15 回で構成される。受講者の大部分は、数学教育を専攻する大学 2 年生で、プログラミングは初心者である。毎年、同じ内容の講義を 2 クラスで実施しており、1 クラス当りの人数は 50 名前後である。条件分岐や反復、配列などの基本的な事項に関する 10 回程度の授業を終えた後に、小テストを実施した。期末試験は、授業で扱った事項全般を含む筆記試験である。全受講者のうち、本実験への協力を表明した学生として、5 年間で合計 363 名分のデータを分析対象とした。

① 基本誤答パターン

表 1 に示す基本誤答パターンについて分析したところ、以下の点が明らかになった。

第 1 に、5 年間の講義の小テストの結果において、誤答として最も多く見られるのは、for ループを解釈しない NFL と呼ぶ誤答パターンであった。全て

の誤答に対する NFL の割合は、年によって多少異なり、最小 26.0%、最大 36.7% だが、概ね 35% 程度であった。参考のために、2015 年度受験者について、誤答パターンに当てはまる解答の割合を表 2 に、ある誤答パターンに半分以上の割合で当てはまる学生の割合を表 3 に示す。

表 2: 2015 年度小テスト受験者の解答全体で基本誤答パターンに一致する割合

パターン名	全体 割合 (%)	誤答中の 割合 (%)
AETI	0.5	1.4
CSOI	0.6	1.7
CVIC	0.3	0.7
CVIC2	2.0	5.7
CVIL	3.8	10.7
CVIL2	0.4	1.2
CVISL	1.6	4.5
NFL	9.2	26.0
RAA	1.4	4.1
いずれかに一致	16.9	47.5
一致無し	18.6	52.5
誤答	35.5	100.0
無解答	11.7	—
正答	52.8	—

表 3: 2015 年度小テスト受験者で基本誤答パターンに一致する学生の割合

パターン名	学生数	全体 割合 (%)
CVIC	1	1.7
CVIC2	8	13.6
CVIL	3	5.1
CVISL	6	10.2
NFL	15	25.4
いずれかに一致	24	40.7

第 2 に、期末試験の成績について、NFL に相当

する誤答が存在する設問のうち半分以上の設問で、NFL に相当する誤答をした学生の平均点は、それ以外の学生の平均点と比較すると、やや低い傾向があった。例として、2015 年度の講義の期末試験の成績データの分布を、箱ひげ図として図 2 に示す。左が期末試験を受験した全 54 名のデータ、真ん中が NFL に相当する誤答が 50% 以上あった学生の 15 名のデータ、右がそれ以外の学生 39 名のデータである。但し、t 検定を行って p 値を求めた結果は、最小 0.00114、最大 0.276 で < 0.05 となったのは 5 年間で 3 回だった。

期末試験素点 (NFL, 0.500)

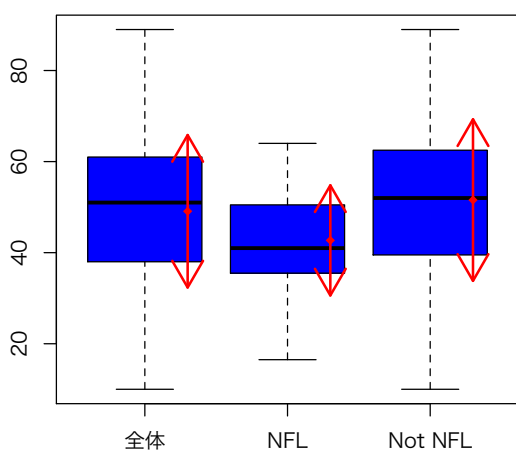


図 2: 2015 年度期末試験の成績の比較

以上を踏まえると、プログラミング学習において、繰り返しを正しく解釈しない学生が常に一定数存在し、このような学生のプログラミング能力は、少なくとも筆記試験で測定する限りにおいて、やや低い傾向があるといえる。繰り返しはプログラムにおいて最も基本的な構成要素の一つであり、これを正しく解釈しないことは、他の構成要素やプログラムに関わる他の概念の習得を阻害している可能性があることが明らかになった。

② 複合誤答パターン

363 名分のデータについて、データマイニング的な手法での誤答パターンの抽出も試みた。具体的には、5 個以上の誤答が含まれている、1% 以上の学生で見られる組み合わせ、基本誤答パターン単独では説明できない、という条件で 22 種類の解答の組み合わせを抽出した。

上記の 22 種類には、NFL と RAA の誤答を 3 個ずつ含む誤答パターンが含まれていた。当該パターンでの解答をした学生の期末試験の成績は、年度によっては有意に低いことがあった。このパターンに該当する学生の一部は、小テスト直後のアンケートにおいて「1..a は数字を列挙しろという意味だと思ったから」といった回答をしており、for ループの記述方法に関する理解が足りなかったことが分かる。

基本誤答パターンはコードの変換として表現していることから、このコードの変換を繰り返し適用して得られたコードの実行結果に一致する複合誤答パターンを考えることもできる。そこで、前述のデータマイニング的手法による解答の組み合わせと、コード変換を 3 回まで適用したコードの実行結果を照合することで、表 1 の CVIL と CVIC2 を順に適用した誤答パターン (以下「CVIL-CVIC2」という) が得られた。CVIL-CVIC2 は「for ループの中の加減算式で、制御変数を用いて計算する」と解釈可能であり、学生のアンケートの回答もこれを裏付けている。但し、当該パターンに当てはまる学生について、期末試験の成績が低い傾向は見られなかった。これは小テスト実施時の単純なミスで、期末試験にまで影響を及ぼすような重大な誤解ではなかったと考えられる。

(2) 既存のオンラインテスト用のツールについて、LTI に準拠したインタフェースを追加開発した。LTI に対応した LMS の一つである canvas (<http://www.canvaslms.com/>) で動作確認をしている。また、誤答パターンに応じたコード変換の仕組みを改良した。ツールを構成するソースコードを整理した上で公開する予定である。

<引用文献>

- ① Mike Lopez, Jacqueline Whalley, Phil Robbins, and Raymond Lister. Relationships between reading, tracing and writing skills in introductory programming. In Proceeding of the Fourth international Workshop on Computing Education Research, ICER '08, pp. 101-112, New York, NY, USA, 2008. ACM.
- ② 関谷 貴之, 山口 和紀, 山本 三雄. 初学者によるプログラムトレーニングにおける誤答に関する分析. 情報教育シンポジウム SSS2012, pp.113-120, 2012.

5. 主な発表論文等

【学会発表】(計 1 件)

- ① SEKIYA, Takayuki, Tracing Quiz Set to Identify Novices' Programming Misconceptions, 13th Koli Calling International Conference on Computing Education Research, 2013 年 11 月 15 日, Koli (Finland)

6. 研究組織

(1) 研究代表者

関谷 貴之 (SEKIYA, Takayuki)
 東京大学・情報基盤センター・助教
 研究者番号: 70323508

(2) 研究分担者

山口 和紀 (YAMAGUCHI, Kazunori)
 東京大学・総合文化研究科・教授
 研究者番号: 80158097