

プログラミング学習における間違えの典型パターンに基づいた学習者分類

Learner classification based on the typical patterns of errors in the programming

都宮 隆之^{*1}, 山岸 秀一^{*1}, 松本 慎平^{*1}

Takayuki TSUMIYA^{*1}, Shuichi YAMAGISHI^{*1}, Shimpei MATSUMOTO^{*1}

^{*1}広島工業大学 情報学部

^{*1}Faculty of Applied Information Science, Hiroshima Institute of Technology

あらまし: プログラミングを不得手とする学習者は各学習者集団の中で一定数存在しているが、従来の教示法に基づく学習では、これらに対して苦手意識を克服することが困難である場合が多い。そこで、プログラミングを苦手とする学生が、どういう理由で不得手と感じているのかを明らかにするために、彼らが犯す典型的な間違えを抽出して学習者を分類する手法を検討した。また、現在プログラミング教育の導入部分で用いられている様々な教育項目が、プログラミング能力の養成にどのように影響を与えるかの分析を行ったので報告する。

キーワード: プログラミング学習, プログラミング素養, 学習者分類, 教育項目

1. はじめに

現在、プログラミングを不得手とする学習者は各学習者集団の中で一定数存在している。そして、従来の教示法に基づく学習では、これらの学習者に対して苦手意識を克服することが困難である場合が多い。従って、プログラミング教育や学習については、より一層の工夫が必要であると考えられる。プログラミングに関連する先行研究としては、例えば、深町は、学生がコンピュータプログラミングを習得する際に陥りやすい間違えを、具体例を挙げながら紹介している⁽¹⁾。また、プログラミングに対する苦手意識を少しでも軽減できるような自学学習環境の構築運用を最終的な目標に設定し、インターネット上で利用可能なプログラミング学習支援システムの開発も行われた⁽²⁾。こうした背景を踏まえて、我々のグループでは、プログラミングを苦手とする学生が、どういう理由でプログラミングを不得手と感じているのかを明らかにするために、簡単なテストによりプログラミング能力が客観的に低い学習者を効果的に抽出する方法を提案した⁽³⁾。これらの研究成果を踏まえ、プログラミング学習支援システムに実装すべき機能や特徴を明確にすることを目的として、プログラミングを不得手とする学生が犯す典型的な間違えを分析し、学習者を分類する手法を検討した⁽⁴⁾。同時に、現在プログラミング教育の導入部分で用いられている様々な教育項目が、プログラミング能力の養成にどのように影響を与えるかの分析を行った。

2. 研究方法

Dehnadi によると、プログラミング学習者の中で、従来手法によるプログラミング教育の効果がある層は、全体の 44%に過ぎず、この割合は、年齢、性別、教育レベルで変わらないとされる。そして Dehnadi らは、プログラミング言語に触れる前にプ

ログラミング素養を予測することができる能力テストを開発した(図 1 参照)⁽⁵⁾。そこで、このテストを各学習者のプログラミング能力の数値化に用い、各人が犯す間違えの程度の指標(誤答率)とする。そして、基礎的なプログラミング素養を問うテストを実施し、この指標を基に各誤解答の程度(ひどさ)を数値化する。さらに、これらから間違えの典型パターンを抽出し、学習者の分類を試みる。一方、プログラミング導入教育で用いられる様々な教育項目の、プログラミング能力養成への効果の度合いの分析には、先に述べた Dehnadi らの能力テスト、フローチャートとトレース、簡単な命令だけを用いたプログラミングの練習、およびプログラミング試験の結果との関係を重回帰分析を用いて調べる。

問題	解答
次の文を読み、右欄から正しい答えを選びチェックしてください。	a と b の新しい値は:
int a = 10;	① a = 10 b = 10
int b = 20;	② a = 30 b = 20
	③ a = 0 b = 10
	④ a = 20 b = 20
a = b;	⑤ a = 0 b = 30
	⑥ a = 10 b = 20
	⑦ a = 20 b = 10
	⑧ a = 20 b = 0
	⑨ a = 10 b = 30
	⑩ a = 30 b = 0
	⑪ a と b はこれ以外の値で、 a = b =

図 1 Dehnadi らの能力テスト (抜粋)

3. 実験と考察

まず、図 2 に示すような基礎的なプログラミングテストを実施し、誤解答の分析を行った。被験者は基礎的なプログラミング教育を一通り終えた学部 2 年生約 100 名である。表 1 に誤解答の程度による学習者の分類結果を示す。表 1 に示す誤解答「x」,

「i+x」, 「num+x」は, 文脈を考えずに直前にある「scanf(“%d”, &x);」に含まれる変数「x」を単純に書き写した, もしくは流用したものと考えられる. これらの誤り程度は 0.06~0.18 である. 一方, 「b[i]」, 「i+2i」といった見当違いの誤りは, 誤り程度が 0.61~0.92 と高くなっている.

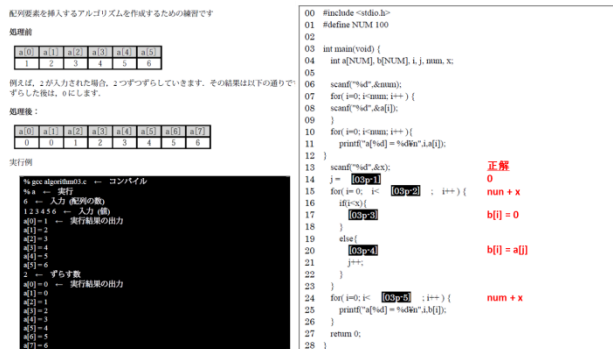


図2 基礎的なプログラミングテスト

このように, 間違えの程度(ひどさ)が高いほど解答の意味が正解から外れる傾向が示された. 同時に, 誤答率により, 直前の行に出てきたものや, 典型的な構文を機械的に当てはめるといった典型的な誤りパターンが分類された.

表1 誤解答の程度による学習者の分類 (抜粋)
問題 03p-1: 正解は「b[i]=0」

解答	誤りの程度	割合[%]
0 (正解)	0	32
1	0.03	17
x	0.15	13
i+x	0.06	3
num+x	0.18	9
b[i]	0.61	3
x-1	0.67	2
i+2i	0.92	1
空欄	1	14

次に, プログラミングを学習する前の導入教育で行った Dehnadi テストの結果, フローチャートとトレースの試験結果, および簡単なプログラミング命令だけによるプログラミング先取り学習の試験結果の4項目をもとに, プログラミング教育を行った後に行う試験の結果を予測する重回帰分析を行った. 被験者は, ほとんどがプログラミング教育を初めて受講する学部1年生約100名である. なお, Dehnadi テストは大学入学直後とプログラミング教育を受ける直前の2回行った. 図3に分析結果を示す. 横軸は実際のプログラミングテストでの得点, 縦軸は予測得点である. 重相関は0.77という高い数値を示し, これらの4項目からプログラミングテスト

結果を十分に予測できることが分かった. さらに, 各4項目の係数は, Dehnadi テストの2要素がともに0.13, フローチャートとトレースが0.43, プログラミング先取り学習が0.20となった. この結果から, プログラミング能力を伸ばすには, フローチャートとトレースをしっかり学習しておくことが, プログラミングルールの先取り学習よりも効果的であることが分かった.

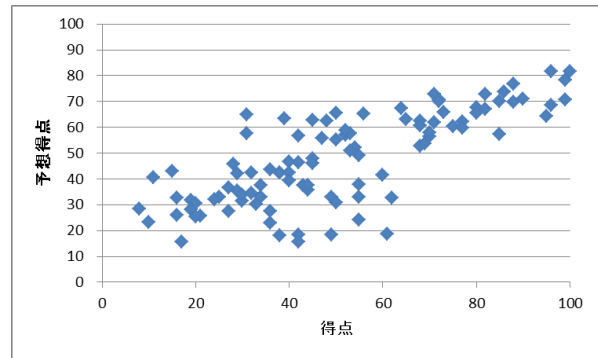


図3 プログラミング試験の重回帰分析

4. おわりに

本研究では, Dehnadi らの能力テストを利用して, プログラミングテストの各解答についての誤程度を数値化することで, 学習者の誤りパターンを分類する手法を提案し, その妥当性を示した. また, プログラミング導入教育時に行われる各項目が, プログラミング能力にどのように影響するかを分析し, フローチャートとトレースが重要な役割を担っていることが分かった. 今後は, プログラミング導入教育の各項目の得点によって学習者を多次元ベクトル化し, グループ分けを行う. そして, 各グループと典型的な誤りパターンの対応を検討し, 誤りを回避するための教育手法を検討する予定である.

参考文献

- (1) 深町秀一: “文系の学生に対するコンピュータプログラミング教育の一考察, 福岡国際大学紀要”, 福岡国際大学紀要, No.23, pp.39-45 (2010)
- (2) 沖中涼, 松本慎平, 山岸秀一, 加島智子: “クライアントサイド動的 Web 技術によるプログラミング学習支援システム的设计”, 平成24年度 JSiSE 学生研究発表会講演論文集 (2013)
- (3) 永吉翔, 山岸秀一, 松本慎平, 沖中涼, 松岡伸彦: “プログラミング教育における学習者の到達度に応じた傾向”, 2012 IEEE SMC Hiroshima Chapter 若手研究会論文集, pp.79-80 (2012)
- (4) 都宮隆之, 山岸秀一, 小野大地, 松本慎平: “プログラミング教育における間違典型パターンに基づいた学習者分類”, 2013 IEEE SMC Hiroshima Chapter 若手研究会論文集, pp.71-72 (2013)
- (5) Saeed Dehnadi, Richard Bornat, “The camel has two humps”, Middlesex University Working Paper (2006)