

プログラム読解学習の狙い及びその効果について ～基本的なデータ依存グラフを把握するスキルの養成に向けて～

On the Aim of Source Code Reading-Based Programming Learning and its Effect -Toward Training the Skill to Grasp the Basic of Data Dependency Graph -

森永 笑子^{*1}, 松本 慎平^{*2}
Shoko MORINAGA^{*1} and Shimpei MATSUMOTO^{*1}

^{*1} 広島工業大学大学院 工学系研究科

^{*1}Graduate School of Science and Technology, Hiroshima Institute of Technology
Email: md18006@cc.it-hiroshima.ac.jp

^{*1} 広島工業大学 情報学部

^{*1}Faculty of Applied Information Science, Hiroshima Institute of Technology
Email: s.matsumoto.gk@cc.it-hiroshima.ac.jp

あらまし: ソースコードを読み、ソースコード実行中・実行後の値を推測する学習は、プログラミングに慣れていない初学者を主として、プログラミング学習者に対する一つの支援法として有効であると考えられる。我々は、プログラミング初学者が躓く要因のひとつとして、読解の際うまくスライスできていないことにあると考えている。よって、プログラムスライスの土台であるデータ依存グラフを的確に、かつ効率良く把握することを狙いとした学習は、プログラミングを苦手とする学習者にとって効果的ではないかと考えられる。そこで本研究では、プログラミングに苦手意識を持つ学習者を支援する前段階として、データ依存グラフを意識することはコードリーディングを適切かつ効率的に行うために有効かどうかを明らかにする。ソースコード読解学習システムを利用して行った実験の結果、データ依存グラフを意識した場合、意識しない場合と比較して、有意に早く読解できるようになることが示唆された。

キーワード: プログラミング, 学習支援, コードリーディング, プログラムスライス, 認知負荷

1. はじめに

プログラミング学習者、とりわけ初学者の認知資源配分を容易にするため、フレーム及び学習活動パターンを制限した学習課題の設計及び開発が進められている。その中のひとつに、ソースコード読解に基づいたプログラミング学習システム(以降、読解学習システム)がある⁽¹⁾。これは、問題として提示されたソースコードを読み、ある変数 x の実行後の値、あるいは、ある変数 x が値 n であるときの他のある変数 y の値、などを答える形式の学習である。読解学習システムは、プログラムを書いて実行するという一般的なコーディング演習と併用して用いるものであり、プログラムに慣れること、効率的にプログラムを読めるようになること(適切にスライシングできるようにすること)、命令を確実に憶えることを目標とした学習である。プログラム中のバグを発見し訂正する際、コードリーディングは特に重要なタスクとなる。誤り訂正を教材とした学習の有効性が示されている^(2,3)、その学習の中でも、読解は最も基本的な技能と位置付けられている。読解は単純で取り組みやすい学習である。よって、読解学習は、バグ発見やデバッグ学習⁽⁴⁾などに接続する前段階の学習として、とりわけプログラミング初学者の一つの学習支援として有効であると考えられる。

我々は、プログラミング初学者が躓く要因のひとつとして、内容理解やデバッグなどの際、本質的に

重要な箇所に意識をうまく集中できていない、すなわち適切にスライスできていないことにあると考えている。非本質的な箇所の読解が余計な認知負荷を生み出し、その結果、円滑な学習を阻害していると考えている。よって、プログラムスライスの土台であるデータ依存グラフを的確に、かつ効率良く把握することを狙いとした学習は、プログラミングを苦手とする学習者にとって効果的であると考えられる。プログラムスライスは効率的な読解に役立つと示されていることから⁽⁵⁾、読解学習はひとつのプログラミング学習の方法として有効であると考えられる。

そこで本研究では、プログラミングに苦手意識を持つ学習者を支援する前段階として、データ依存グラフを意識することはコードリーディングを適切かつ効率的に行うために有効かどうかを明らかにする。ソースコード読解学習システムを利用して行った実験の結果、データ依存グラフを意識した場合、意識しない場合と比較して、有意に早く読解できるようになることが示唆された。

2. 読解学習システム

ソースコード読解学習システムは、C 言語の読解学習を対象としたものであり、ソースコードを自動生成できる。また、HTML5 に準拠した Web ブラウザさえあれば、システムの全機能を利用可能である。システムは Apache 2.4.7 で動作しており、問題提示

には JavaScript ライブラリである jquery 1.7.2, 問題データ管理には MySQL 5.6.16, Web アプリケーションは php 5.5.9 を用いている. 教授者は, 問題の自動生成機能を用いることで, 読解教材を自動的に作成できる. 学習者のメインユースケースである演習画面を図 1 に示す.

従来のシステムは, 変数や命令の数など問題生成に関する制約を設定した後, 任意の規則に基づいてソースコードを生成する機能のみ有している. そこで本研究では, 従来の読解学習システムを拡張し, データ依存関係を反復学習できるような機能を追加した. これにより, データ依存関係を把握しながら効率良く読解する力の獲得を目指した学習を実現できるようになった.

3. 実験

データ依存グラフを意識することはコードリーディングを適切かつ効率的に行うために有効かどうかを明らかにするため, C 言語及び Java 言語の基礎やアルゴリズムの基礎といったプログラミング関連科目の基本を既に習得済みの大学 3, 4 年生に対して評価実験を行った. 実験は, 2018 年 2 月 2 日からの 3 週間の間に 11 名に行った. 本実験で出題されたソースコードは, インクリメントや複合代入演算子といった命令を用いていない. また, 用いられる演算子は四則演算と余剰のみである. このような条件で生成されたソースコードを課題として提示し, 指定した変数の実行後の値の回答を被験者に求めた.

実験の流れを示す. まず, 時間無制限のプレテストを行い, 被験者各自のやり方で読解を行った. 次に, データ依存グラフと, それに基づいて読解する戦略を教授し, データ依存に沿って読むことに慣れるトレーニングを 30 分行った. トレーニング後, 時間無制限のポストテストを行った. プレ・ポストテストでは, データ依存関係のみを持った代入文のみのソースコードを 5 問, if 文を含むソースコードを 5 問, while 文や for 文を含むソースコードを 5 問, if 文と while(for)文を両方含む 5 問の計 20 問を被験者に出題した. なお, プレ・ポストテストの難易度は同程度とした. トレーニングでは, 4 行のソースコードで可能な 20 種類のデータ依存グラフに従って計 20 問の問題を生成し, 課題として被験者に提示した.

図 2 に実験結果を示す. 図 2 により, 事前(プレテスト), 事後(ポストテスト)の読解時間の間には, Welch の t 検定により有意な差($p < 0.05$)が確認された. なお, プレ・ポストテストの正答率には, 有意な差は確認されなかった. このことから, データ依存グラフを意識することで, 被験者は効率よく読解できるようになったと考えられる.

4. おわりに

本研究では, データ依存グラフを意識することはコードリーディングを適切かつ効率的に行うために有効かどうかを調査した. 実験の結果, データ依存

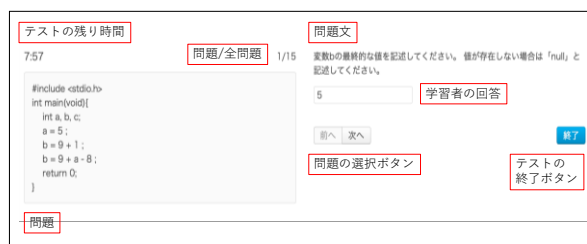


図 1 テスト画面

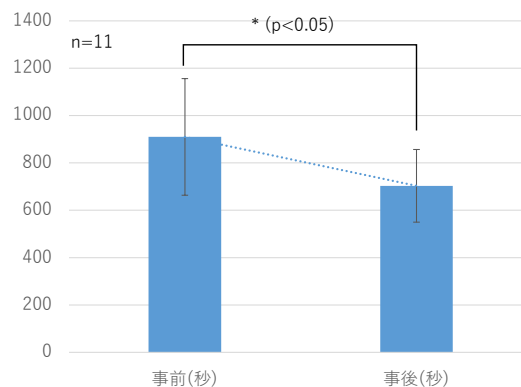


図 2 実験結果 (エラーバー: SD)

グラフを意識した場合, 意識しない場合と比較して, 有意に早く読解できるようになることが示唆された. 今後, 「プログラミングに苦手意識を感じる学習者はうまくプログラムスライスできていない」といった仮説の検証が課題となる. プログラミングに苦手意識を感じる学習者の場合, 通常の被験者よりも改善の幅が大きければ, その仮説はより確かなものであると考えることができる.

謝辞

本研究は, 独立行政法人日本学術振興会科学研究費助成事業(基盤研究(C)16K01147, 17K01164)の助成を受けて実施した成果の一部である. ここに記して謝意を表します.

参考文献

- (1) K. Okimoto, S. Matsumoto, S. Yamagishi, T. Kashima, Developing a source code reading tutorial system and analyzing its learning log data with multiple classification analysis, *Journal of Artificial Life and Robotics*, Volume 22, Issue 2, pp 227-237 (2017)
- (2) 蜂巢吉成, 吉田敦, プログラミング学習における誤り訂正問題の自動生成方法の提案, *信学技報*, vol.112, No.457, pp.63-68 (2013).
- (3) 長瀧寛之, 伊藤亮太, 大下福仁, 角川裕次, 増澤利光, アルゴリズム学習における間違い探し形式の演習課題を自動生成する手法の提案と評価, *情報処理学会論文誌*, Vol.49, No.10, pp.3366-3376 (2008).
- (4) 江木鶴子, プログラミング初心者を対象にしたデバッグ戦術の学習支援に関する研究, *九州工業大学博士論文*, 甲第 225 号 (2009).
- (5) 西松顯, 西江圭介, 楠本真二, 井上克郎, フォールト位置特定におけるプログラムスライスの実験的評価, *信学論 D-I*, Vol. J82-D-I, No.11, pp.1336-1344 (1999)