

プログラミング教育におけるデザインパターンの判別 Discrimination of Design Patterns in Programming Education

伊東 大輝[†]
Hiroki Ito

島川 博光[†]
Hiromitsu Shimakawa

1. はじめに

情報教育を目的とした教育機関では、プログラミングの知識が熱心に指導されている。文法の知識は、プログラミング能力の育成において重要である。しかし、プログラミング能力は文法の知識だけでは成り立たず、いざプログラムを作るとなると途端に手が止まってしまう学生が多い。このような学習者の多くは、文法の知識はあるが、与えられた問題文を実現するための知識の結びつけができていないと考えられる。

構造化プログラミングによれば、学習者は大きな問題を部分問題や知識に分割し、それらを組み立ててプログラムを作り上げる。本論文では、学習者が知識を結びつけ、論理の組み立てができることをスキーマが形成されていると表現する。プログラミングの知識が習得できていても、論理を組み立てられずソースコードが書けない学習者は、プログラミング能力があるとは言えない。指導者は、このようなスキーマが形成されていない学習者を早期に発見し、対策を講じる必要がある。本論文では、学習者に PAD 表現を用いた並び替え問題を解かせ、その学習行動から学習者のスキーマ構築度を推定する手法を提案する。本手法は、分析結果から学習者の不理解の原因を特定する。指導者は、特定された原因に応じた指導ができる。

2. プログラミング教育の課題

2.1 プログラミング教育を支援する関連研究

Mysore ら [1] は、学習者が苦勞している箇所を特定できる Web システム Porta を提案した。Porta は、本来複雑に要因が絡み合っているはずの理解度を、苦勞したという点にのみ着目して推定しており、学習者の能力や成長を考慮できていない。Guo ら [2] は、リアルタイムで一対多のプログラミング学習を支援するインタフェースとその実装 Codeopticon を提案した。しかし Codeopticon は、インストラクターの質に依存し、インストラクターに大きな負担を強いる。これら多くの既存研究は、知識の側面にばかり着目しており、論理に基づく理解度推定を行っていない。伝統的なプログラミング教育の形式の一つとして、Parsons ら [3] の提案した Parson's programming puzzle がある。これは初学者が取り組みやすいツールとして紹介されている。しかし、このツールは論理上の構成要素の習得に着目しているものの、理解度を推定するものではない。

2.2 PAD によるプログラム図式表現方法

私たちが「電車に乗る」という問題に取り組む時、「切符を買う」、「改札を通り電車に乗る」、「目的地で電車を降りる」といった知識の集合とそれらの組み合わせ方・筋書きが頭の中に存在している。本論文では、このような知識の集合と知識の組み合わせ方をスキーマと呼ぶ。

プログラミング学習では、学習者の理解度を測るため

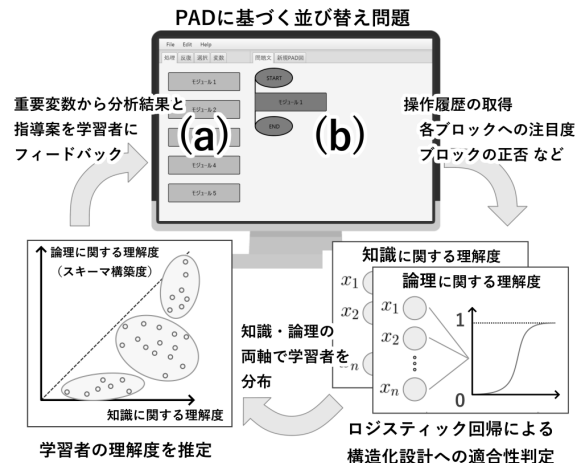


図 1: スキーマ構築度推定モデルを用いた教育支援の概要

に全文記述問題や空欄補充問題が用いられることが多い。全文記述問題は、問題文や条件を読み、それらを満たすソースコードをすべて記述する。指導者は採点時に学習者の理解度を細部まで判定できるが、回答の多様性があり採点が難しい。空欄に正しく動作するコードを記述する空欄補充問題は、回答部分が限定されているので、空欄の正誤から理解度の有無を判別しやすい [4]。しかしそこでは、知識を紐付け、論理を組み立てる能力が考慮されていない。

本手法では、学習者は PAD [5] を用いてコードブロックの並び替えを行う。PAD は、横軸方向に複雑さ、縦軸方向に処理の大きさを表し、段階的詳細化法に適している。また PAD は、修得が容易であり、基本制御が接続・選択・反復・処理に限定されている。PAD による並び替え問題は、全文記述問題や空欄補充問題と異なり、知識ではなくそれらを組み立てるスキーマの適用と形成に作業に集中できるため、本研究の手法に適している。

3. スキーマ有無の判別法

3.1 スキーマ有無判別による教育支援

本研究では、学習者がプログラミングの並び替え問題を解く際の行動から、学習者に適切なスキーマが形成されているかを判別する方法を提案する。図 1 に手法の概要を図示する。本手法は、スキーマの形成の判別結果から、学習者の不理解の点をも明らかにすることができる。

構造化プログラミングは、大きな問題を部分問題に段階的に区切る。この区切りの最小単位は、学習者が授業などで学習した制御構文の文法や関数ライブラリなどの知識である。学習者は、これらの知識の断片を組み合わせ、より複雑なプログラムを作り上げる。本論文では、この知識の組み合わせの作業を論理の組み立てという。

本手法は、スキーマ有無とその要因を判別するために、複数の学習者から学習行動のデータセットを収集する。

[†]立命館大学情報理工学部

目的変数として、実際にスキーマが形成されているかは、事後のインタビューからヒトが判断する。本手法は、複数の学習者の学習行動のデータセットから、スキーマ構築度を知識の面、論理の面の理解度の両軸からロジスティック回帰により分類する。指導者は、ロジスティック回帰での重みが高い変数から、スキーマ構築度に寄与したプログラミング要素を特定できる。この分類器を新規学習者に適用すると、新規学習者に足りないプログラミングの要素が特定できる。

3.2 PAD に基づく並び替え問題

本手法は、スキーマ形成有無の特定のために学習行動を収集する。学習行動の記録には、図 1 の Windows 上で動作する PAD エディタを用いる。図 1 a は、反復や分岐などの制御命令やモジュールなどのブロックを選択するための領域で、図 1 b は、PAD 組み立てのキャンバス領域と、問題文の表示を切り替えるタブである。

学習者は、図 1 a の領域から、この問題を解決するのに必要であると考えられるモジュールや制御命令などのブロックを選ぶ。学習者は、選択したブロックを、図 1 b の PAD キャンバス領域へ移動させる。迷った学習者は、この学習行動の中で特別な行動をとる。本手法では、この迷った行動に着目する。

3.3 操作記録の取得

学習行動として以下を記録する。

○各ブロックの注目度

ブロックの説明を見た回数とブロックを移動させた回数を収集する。学習者は、与えられたブロックをマウスオーバーすると説明を見ることができる。学習者がブロックの説明を見た時間は、そのブロックに対する知識の有無に関連すると考えられる。あるブロックについての操作が多い学習者は、そのブロックと他のブロックの知識との組み立てができいていないと考えられる。

○ブロックに関する正否とその時間的推移

各ブロックの正否と全ブロックの正答率を収集する。学習者は、ブロックを論理的に正しい位置に置く。本手法では、ブロックの前後に繋がる要素が正しい時、論理的に正しい位置とする。ある問題について、論理的に正しいブロックの組み合わせが一部でもできていれば、その学習者はその部分については理解をしている可能性がある。全体のブロックのうち論理的に正しい位置に置かれたブロックの割合を時間順に観測し、正答率が急激に上がっている学習者は、この問題を解いている間にスキーマが形成された可能性があると考えられる。

○作図に費やした時間

PAD エディタは、問題文と作図画面が同一タブになっている。そのため学習者は、問題文と作図画面を交互に見なければならない。作図画面に集中できている学習者は、既にスキーマが形成されており問題文をみてすぐにプログラムの設計ができると考えられる。作図が始まらない学習者は、スキーマが形成されておらず、どのようにプログラムを設計すべきかを見通せていないと考えられる。

3.4 スキーマ有無とその要因の同定

並び替え問題の学習行動からスキーマ有無の判別を行うため、ロジスティック回帰により、知識に関する理解

度と論理に関する理解度のモデルを生成する。

本研究では、知識と論理の両方を持っている場合、スキーマが構築されていると同定する。本研究では、スキーマを形成する要因を明らかにすることが目的であるため、モデルは並び替え問題ごとに生成する必要がある。生成されたモデルを新規学習者に適用すると、学習者のスキーマの有無が判別できる。生成されたモデルは、スキーマ有無を特定するための、その問題に特有な重要変数をもつ。スキーマ有無を特定するためのモデルは、前項であげた説明変数が複雑に絡み合っている。ブロックの説明を見た回数は知識に関する理解度に寄与し、操作した回数は論理に関する理解度に寄与すると考えられる。

全ブロックの正答率が低い場合でも、一部のスキーマは形成されている可能性をブロック単位の成否から判明する。また作図に時間を割けなかった場合には、知識による要因と論理による要因のどちらの可能性も考えられる。もしブロックの説明を見た回数が多ければ、学習者に十分な知識が身につけていないことが裏付けられる。このように、スキーマ形成がないと判別された新規学習者の重要変数を考察すれば、足りないプログラミングの要素が特定できる。さらに、学習者の理解度は、モデルの応答変数から、知識と論理の両軸で表した空間にプロットされる。よって指導者は、これらの情報より対策を講じることができる。

4. おわりに

本論文では、プログラミング教育の現場で学習者のスキーマ構築度を推定する手法を提案した。本手法は、PAD を用いた並び替え問題への取り組みから、学習者のスキーマ構築度とその要因を明らかにする。指導者は、理解に躓いている学習者を早期に発見することができ、より適切な対策を講じることができる。今後は、本手法の有効性を実験により検証する。

参考文献

- [1] Mysore, et.al., Porta: Profiling software tutorials using operating-system-wide activity tracing, Proc. of ACM symposium on User Interface Software and Technology, pp. 201-212, 2018
- [2] P. J. Guo. Codeopticon: Real-time, one-to-many human tutoring for computer programming. Proc. of ACM Symposium on User Interface Software and Technology, pp. 599-608. 2015.
- [3] D. Parsons, et.al., Parson's programming puzzles: a fun and effective learning tool for first programming courses. Proc. of Australasian Conference on Computing Education, Vol. 52, pp. 157-163. 2006.
- [4] So Asai, et.al., Predicting cognitive load in acquisition of programming abilities. Proc. of International Journal of Electric and Computer Engineering(IJECE), Vol. 9, No. 4, pp. 3262-3271, 2019.
- [5] 二村良彦, 他. Pad (problem analysis diagram) によるプログラムの設計および作成. 情報処理学会論文誌, Vol. 21, No. 4, pp. 259-267, 1980.