

# コード間の関係記述モデルの提案

渡辺 圭祐<sup>†</sup> 東本 崇仁<sup>††</sup> 藤森 進<sup>†††</sup> 赤倉 貴子<sup>†††</sup>

<sup>†</sup> 東京理科大学大学院工学研究科

<sup>††</sup> 東京工芸大学工学部

<sup>†††</sup> 東京理科大学工学部

## 1. はじめに

プログラミング学習の方法として、プログラムを作るばかりではなく、プログラムを読むという学習法が提案されている。著者らは、プログラムを読む学習のうち「意味理解学習」において段階的に考えるプロセスを「段階的抽象化」と定義し、段階的抽象化プロセスにより「意味理解学習」を支援するシステムを開発した[1]。しかし、このシステムにおける学習者へのフィードバックは、正誤情報のみを与えるにとどまっている。本来、誤りの発生する箇所は学習者ごとに異なるため、学習者に適応的なフィードバックを付与することが望ましい。より詳細なフィードバックを付与するためには、問題間の関係性を記述したモデルが必要となる。モデルの記述により、補助問題の提示や発展問題の出題などを可能にしたより質の高い学習支援システムの開発が可能になると考える。本稿では、段階的抽象化によるプログラムを読む学習を支援するシステムのフィードバックの拡充のために必要となる、問題としてのプログラムコード間の関係を記述する方法について議論する。

## 2. コード間の関係性を記述する方法

前述した目的の達成のためには、問題間の難易度の順序付け、及び隣接する問題の決定が必要となる。これらを達成するために、問題としてのプログラムコードの関係性をグラフにより記述することを目的とする。グラフ作成の際には、学習者に対する補助問題の提示を考え、難易度に基づいた構造化を行う。そこで、ある問題に対して簡単な問題として、(1)部分化、(2)特殊化、された2つの問題を考える。

図1は、代入式の配列のインデックスの値が可変である問題と、それを特殊化した例である。図中左のコードは配列のインデックスが変数  $i$  と  $j$  であり、その値に制約はない一般的な値であるが、右のコードは変数の値を特殊化している。つまり、特殊化前のコードに比べて限定的な動作をするコードとなる。学習者が頭の中で追わなければならないパラメータが少なくなるから、特殊化により問題は簡単になると考えることができる。

部分化されると、if文やfor文などの有意味な単位の部品から構成されるコード全体を、各部分に分割することとなる。図2の例では、if文とSwapから構成されるコードがSwapのみのコードとなり、その後Swapを構成する要素となる代入を意味するコードへと部分化されている。また、部分化された問題はさらに部分化することができ、最終的にコードとして有意味な1行となる。

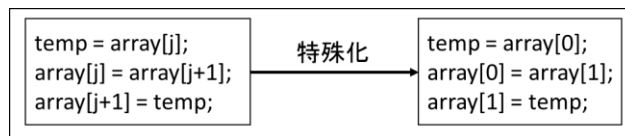


図1 特殊化の例

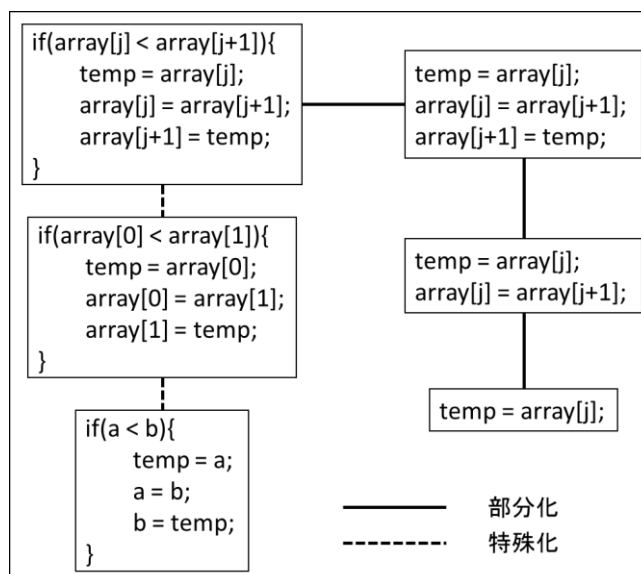


図2 モデルの例

## 3. まとめと今後の課題

段階的抽象化によるプログラムを読む学習を支援するシステムにおけるフィードバックの充実を目的とし、その目的の達成のために必要となる、グラフを用いたコード間の関係性の記述の方法を議論した。今後の課題として問題間の差分について検討することが挙げられる。

## 謝辞

本研究の一部は、科学技術機構・社会技術研究開発センター「問題解決型サービス科学研究開発プログラム」の研究開発プロジェクト「高等教育を対象とした提供者のコンピテンシーと受給者のリテラシーの向上による競争的価値の実現方法の開発」の成果によるものである。

## 参考文献

- [1] Keisuke Watanabe, Takahito Tomoto, Takako Akakura, “Development of a Learning Support System for Reading Source Code by Stepwise Abstraction,” Human Interface and the Management of Information, Information and Knowledge in Context, LNCS 9173, pp.387-394, 2015