

多様な知識への注目を促す プログラミング再学習支援環境に向けて

土屋孝文*1・西岡大吾

Email: tsuchiya@sist.chukyo-u.ac.jp

*1: 中京大学工学部

◎Key Words プログラミング, 理解方略, 学習支援

1. はじめに — 例題プログラムからの学習

本研究は情報系基礎科目「アルゴリズムとデータ構造」を対象に、学習支援環境の開発と運用を行っている。この実践では、先輩にあたる学生が自分自身の学習経験に基づき後輩たちの学習支援環境設計に関わるサイクルを継続している。

この科目の学習には3種の領域知識、すなわち、対象問題のアルゴリズムの理解、アルゴリズムに対応するコーディング（プログラム生成）、プログラムの実行過程とアルゴリズムとの対応（プログラム解釈）が相互に関係している。本研究は各領域について、学習者に応じた支援環境の設計を検討してきた。

アルゴリズムの理解に関する環境については、具体的な解法事例に対する学習者自身の手続き説明（操作のなぞり）から始め、協調的活動を利用して集団中の解や思考を比較の後、規範的な解答との対照までを基本的な設計フレームワークとし、対象問題に応じた環境の提供と運用を行っている。

アルゴリズムの理解と比べて、プログラムの生成や解釈には学習者に一定の難しさが観察される。プログラム生成の支援には、基本的な問題処理知識に対応する例題プログラム集を準備し、一旦これらを学習した後の学習者に、課題に応じた適切な例題の利用を促す環境の運用を試みている⁹⁾。表1はバブルソートなどの基本ソートをプログラムするレベルまでに必要と考えられる9つのカテゴリと例題である。

表1 基本例題集

	カテゴリ	例題1	例題2
1	入出力	おうむ返し	配列格納
2	条件分岐	偶数/奇数判定	月 ⇒ 季節
3	ループ	N個合計	N個/終了キーまで合計
4	配列処理	要素合計	連続要素縮約
5	文字列	長さ	1文字検索
6	二重ループ	九九表	一次元配列 ▽型出力
7	その他 (基本処理)	配列要素交換 (swap)	乱数ライブラリ
8	関数	sum 関数	配列要素の合計
9	再帰	階乗	二分探索

これまでの運用からは、課題中に例題を参照し、その有用性に肯定的な評価を行う学習者がみられる一方で、学習済の例題であってもそこから有用な知識をうまく活用できない、いわば学習が不安定あるいは不完全な状態にある学習者も多い。そのような学習者に対する例題プログラム提供法について、適切な（再）学習が促されるように再設計の工夫が求められる。

江川ら(2003)は、初学者のプログラミング学習過程にみられる不安定さや不完全さについて、二重ループプログラムに対する不十分な説明事例を基に考察している¹⁾。彼らは、不完全な学習状態について、より基本的な知識を積み上げていく過程に起こる準備不足だけではなく、教授者側が見落としがちな、学習者の暗黙的理解方略の結果ととらえている。そのような方略には、プログラムの内部処理制御に立ち入らず、プログラムの入出力の理解にとどまる方略や、プログラム解釈に必須なプログラム実行過程を推論せず、構文パターンだけの理解にとどまる方略があげられている。

本稿は、（再）学習時にプログラムの実行過程への注目を促すため、実行順序を対話的に確認する機能を追加した例題プログラム提供法の改良とその運用事例を報告する。

2. 確認知識の切り替え

表1の例題集に、より入門的な6つの例題（if, switch, while, for, 配列, 関数）を追加した。このような基本知識は、長期休暇明けなど、一時的に不安定になったプログラム知識を確認する手がかりに有用と考えられる。入門的な知識の確認については、85.7% (N=91) のユーザより有用との評価を得た²⁾。

```

for
  構文 実行 流れ

#include <stdio.h>
int main(void) {
    int i;
    for (i = 0; i <= 5; i++) {
        printf("%3d ", i);
    }
    return 0;
}

実行結果
0 1 2 3 4
  
```

図1 for 文例題 変数値に応じた実行結果の確認

現在の for 構文の例題ウェブページでは (A) 構文の知識、(B) 変数値に応じた実行結果⁽⁴⁾、(C) プログラムの実行過程と、確認すべき知識を切り替えることができる。図 1 は、変数値設定後、JavaScript のイベント処理を用いて、ウェブページ上のプログラムと同じ計算を行い、プリント文の出力結果を生成、表示する。

図 2 は、プログラムの実行順序を選択、確認していくクイズページである。変数値と標準出力を左のサブウィンドウに順次、表示する。対話的なクイズ機能を用いて 6 例題を (再) 学習する活動を行った結果、93.8% (N=32) のユーザより有用との評価を得た。



図 2 for 文例題 実行順序の確認クイズ

この活動はコンピュータ演習室で行われており、回答の誤りについて、表示結果を共有しながら周りとの説明しあう様子が多くみられた。確認された知識の有無については、ユーザ全員が有と回答し、34.3% (N=32) が複数の確認を回答した。

3. プログラムの理解方略

エキスパートのプログラム理解を支える主な知識処理には、プログラムに利用されている意味的処理単位 (基本的な問題処理知識、スキーマ) の同定、対象問題とプログラムとの対応に関する推論、および実行過程の推論がとりあげられてきた⁽²⁾。

この分類から現在の例題プログラムの特徴を整理してみると、まず例題プログラムは、初学者にとって具体的でわかりやすい対象問題に関する、適当なサイズの解法プログラムと考えられる。初学者には、例題プログラムの適切な理解を通して、プログラム中に使用されている基本的な問題処理知識 (定型的な処理パターン) の学習が期待されている。江川ら (2003) の指摘は、実際の初学者の例題プログラムの理解が、わかりやすい対象問題の読み取りや入出力に関する推論にとどまり、プログラム自身の内部処理や実行過程の読み取りまで進まない場合と考えられる。

ここでプログラムが何をするのかに注目する表層的な読みを「物語読み」、プログラム内部の処理や制御および実行過程を追跡する読みを「トレース読み」とよぶことにすると、江川ら (2003) の指摘は、初学者によっては「物語読み」のしやすい例題について十分な「トレース読み」を行わない心配がある、と表現できよう。図 2 のような実行順序確認クイズは、「5 回プリントす

るプログラム」という物語読みから、そのような出力を行う制御に関するトレース読みを促す方法の 1 つで、一定数の初学者が誤解について自分 (たち) で修正できた事例と考えられる。

図 3 は、トレース読みをしなければ printf 文の出力を予想できない、いいかえると物語読みのできないプログラム例である⁽¹⁾。長期休暇明けの初学者の出力正答率は、ケアレスミスも考慮しても予想以上に低い場合がある。そこで実行順序確認クイズによる再学習後に、誤答に関係した知識や推論について選択肢による回答を求めたところ、基本的なループ処理に関する不安定な理解への気づきや自己修正の様子が確認された (N=43)。

問2

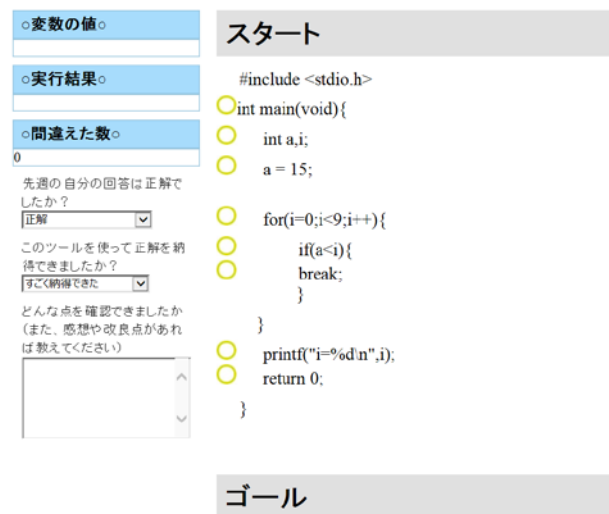


図 3 トレース読みによる知識確認

4. おわりに

例題プログラムからの学習が期待される知識は、定型的な変数使用法やプログラム生成方略など多様である。初学者の暗黙的な理解方略を考慮して、対象知識へ適切な注目を促す再学習支援法の検討が重要である。

エキスパートは、トップダウンな物語読みとボトムアップなトレース読みを切り替えながら、プログラム中の変数や処理の意味をアルゴリズムに対応づける読み (「アルゴリズム読み」とよぶ) ができる。初学者のアルゴリズム理解 (操作のなぞり) を、プログラムのアルゴリズム読みへつなげる支援が次の課題である。

参考文献

- (1) 江川紘美, 三宅芳雄: "詳細な観察に基づくプログラミング学習過程の研究", 中京大学人工知能高等研究所ニュース (IASAI News), 12, pp.10-15 (2003).
- (2) J-M. Hoc, T. R. G. Green, R. Samurcay and D. J. Gilmore (Eds): Psychology of Programming. Academic Press (1990).
- (3) 土屋孝文, 高井峻太: "例題を手がかりにしたプログラミング基本知識の準備支援環境", PCカンファレンス論文集, pp.61-62 (2017).
- (4) 占部弘治: "プログラミング科目の理解を助けるインタラクティブ教材の検討", PCカンファレンス論文集, pp.63-64 (2016).