

解答プログラムからの学習支援

—多視点からのより深い読みを促すために

土屋孝文*1・児玉一朗・池内愁・横井将幸

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

*1: 中京大学工学部

◎Key Words アルゴリズム, プログラミング, 学習支援

1. はじめに — 解答プログラムからの学習

本研究は、C 言語の入門科目に続く、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。

プログラミング演習の側面から見ると、現在の授業は大まかに 3 種のフェーズ、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（解答例となるプログラムの理解）の順に進行する。このうちコーディングは、プログラミング演習課題となる。学習者には解答例プログラムからの知識の獲得や修正が期待される。

新しい知識の獲得にあたるアルゴリズムの理解に比べて、入門科目を通して得られた知識の運用にあたる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。ここには学習者の既有知識の差、プログラム生成方略や解釈方略の多様性、実際の知識運用の不安定さなどを考慮した再学習の支援が必要である⁽¹⁾。

本稿ではプログラミングに関する 2 つのフェーズのうち、解答例プログラムからの再学習に関する支援について報告する。1 つの解答プログラムから学習できる知識や情報は、実行過程、実行過程とアルゴリズムとの対応づけ、プログラム生成方略、変数や制御構造の定型的な使用方法など多種多様である。プログラムの初學者には、解答プログラムだけではなく、複数の視点からの読みと学習を促す支援の提供が重要と考えられる。以下では、そのような試みとして、プログラムの実行過程とアルゴリズムとの対応づけの支援、およびプログラム生成過程の推定に関する支援を報告する。また、より深い読みへの方向づけのため、プログラムへの説明的コメント文作成を促すガイドライン案を報告する。

2. 実行過程とアルゴリズムの対応づけ

記号的情報表現にあたるデータ構造や擬似コードではなく、トランプや棒グラフなど直観的な図的情報表現を用いて具体的な操作系列列と説明を示し、アルゴリズムの理解を支援する学習素材は多い⁽²⁾。対象のアルゴリズムを理解すれば、図的情報表現による操作手続きを適切になぞれるようになる。この理解を基に、プログラムと実行過程を、図的情報表現と操作に対応づけることで、プログラムの理解を支援するツールが考えられる⁽³⁾。

初學者によるプログラムの読みには、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、構文パターンへの注目にとどまる方略がみられる。そのため支援ツールには、重要な変数の値の変化やプログラム実行順を確認し、学習者にプログラムのトレースにあたる読みを促すための工夫が必要である。

以上を背景に作成した Web ページの例（二分探索）を図 1 に示す。ページは先行フェーズで確認した図的表現の変化と対応づけながら、中央のフレームでプログラム実行過程をトレースする。左のフレームには重要な変数の変化を表示する。

二重ループや基本再帰を含んだ 6 つの基本アルゴリズムに関する実行過程確認機能について、65.6% (N=61) のユーザーから肯定的な評価が得られた。今後は言語的な説明や対話的な知識確認機能の追加を検討している。

The screenshot shows a web interface for a binary search algorithm. It includes a code editor with the following C code:

```
int binary_search(int key) {
    int low = 0;
    int high = N - 1;
    int mid;

    while (low <= high) {
        mid = (low + high) / 2;
        if (s[mid] == key) {
            return (mid);
        }
        if (s[mid] > key)
            high = mid - 1;
        else
            low = mid + 1;
    }
    return -1;
}
```

On the right, a diagram shows an array of numbers: 1, 3, 4, 8, 13, 14, 18, 20, 21, 25. A search key of 8 is shown, and the 'mid' index is 3, pointing to the value 8. The 'low' pointer is at index 0 and the 'high' pointer is at index 9.

図 1 解答プログラム実行過程への注目を促すページ

3. プログラム生成過程の推論支援

一般にプログラムの基本例題は一度に全体が学習者に示される。制御の基本構文を学ぶには適当と考えられるが、プログラムを部分的に追加し生成していくコーディング過程を学ぶためには、学習者自身が、いわゆる写経的練習を通してプログラムの意味的な単位（粒度）や標準的な生成順序を推論しなければならない。

そこで解答プログラムの標準的なフローチャート表現を基に、その 1 手続き（ユニット）を粒度とし、全ユニットの具体化順序を経験的に求めたスクリプトを準備した。このスクリプトに従って、プログラムを部分的に示して

いく Web ページを作成、運用した。図 2 は、24 ユニットに分割した二分探索のプログラムについて、20 番目のユニットを表示しているページで、次のフレームで追加されるユニットの推論を促している。

現状の実装は 1 つの定型的な生成例を示すだけであるが、13 の解答プログラムを対象にページを運用したところ、コーディング過程の学習に対する有用性について、72.2% (N=54) のユーザから肯定的な評価が得られた。

```

STEP 2.2 二分探索 ( 20 / 24 )
Prev Next
31: void print_data(void) {
32:     int i;
33: }
34: for(i = 0; i < N; i++) {
35:     printf("s[%d] = %d\n", i, s[i]);
36: }
37: }
38:
39: int binary_search(int key) {
40:     int low = 0;
41:     int high = N - 1;
42:     int mid;
43:
44:     while(low <= high) {
45:         mid = (low + high) / 2;
46:         printf("low = %2d, high = %2d, mid = %2d\n", low, high, mid);
47:
48:         if(s[mid] == key) {
49:             return (mid);
50:         }
51:
52:         if(s[mid] > key) { /* 探索キーより中間の配列の値が大きい場合 */
53:         }
54:     }
55: }

```

図 2 解答プログラムの生成過程への注目を促すページ

4. プログラムを説明するためのガイドライン

プログラムのより深い読みをもたすには、変数の役割やアルゴリズムとの対応などを言語化し、プログラムに説明的なコメント文を付加する活動が考えられる。一般にコメント文に関する参考書は熟達者に向けた解説が多く、プログラムの意味を繰り返して表現する説明的なコメントは推奨されない⁴⁾。一方で、指導をうけない初心者の説明的なコメントには、プログラムを自然言語に置き換えだけの表面的な表現が多く見られる。

そこで、代表的な参考書中の学習者向けプログラムコメント例を参考に、初学者が解答プログラムを読んで説明的なコメントを付加しようとする際のガイドラインを検討した。現在のガイドライン案はコメントの一般的な方針、形式に関する方針、内容に関する方針の 3 つに大別され、全体で 13 項目から構成される。以下に、それぞれのガイドラインを示す。

一般的な方針

- G1: 読み手に説明するつもりでコメント文をつけよう
- G2: プログラムを見ただけではすぐにわからない、でも、言葉で説明しておくでプログラム理解の助けになると思う部分にコメントしよう
- G3: 説明しなくてもわかるだろうと思われる部分はコメントしなくてよい
- G4: 自分もはっきりわかっていない部分があれば、どう考えているか、あるいは、何がはっきりしないかなどをコ

メントしておく

G5: 良いコメントを真似るだけではなくて、自分らしい表現もしてみよう

形式に関する方針

- S1: 行の上を書くブロックコメントと、プログラムの後ろに書く行末コメントを使い分けよう
- S2: コメントの中の空白や行揃えなど、できるだけスタイルを統一しよう
- S3: 簡潔で分かりやすい表現を心がけよう

内容に関する方針

- C1: 変数やデータ構造が、どんな情報を表現しているか
- C2: 定数が、どんな情報を表現しているか
- C3: 変数が、プログラムが表現している手続きの中で、どのような役割をしているか
- C4: ひと固まりのプログラム (ブロック) や関数が表現している手続きの概要
- C5: 1 行程度のプログラムが表現している手続き

解答プログラムに対して、内容に関する項目番号を対応づけながらコメント文を追加する Web ページを作成した。さらに、教員が作成したコメント例と対照して、各コメントについて選択肢による振り返り評価を行った。

配列の初期化に続き二重ループを用いて配列を繰り返し表示するプログラム (全 12 行) について、教員が追加した説明コメント数 7 に対し、ユーザの平均コメント数は 4.6 (N=69) だった。コメントに対する項目番号の選択には一定数の誤りが見られるため (正答 71.9%)、具体例を用いた事前の練習が必要と考えられる。ガイドラインの有用性については、97.3% (N=39) のユーザから肯定的な評価が得られており、ガイドラインを繰り返し利用して項目の調整を行う予定である。

5. おわりに

現在の支援 Web ページについては、より自動的な生成方法を検討しているところである。また、ページには対話的な機能の追加が重要である。

プログラミングの初学者は、知識が不完全なまま、運用も不安定ながら、次の学習場面へ進むように観察される。再学習時の支援や方向付けという観点から、プログラミングの学習支援環境を整理していく必要があると考えられる。

参考文献

- (1) 土屋孝文・西岡大吾: “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”, 2018 PC カンファレンス論文集, pp.21-22 (2018).
- (2) 矢沢久雄: “トランプでおぼえるアルゴリズム”, 日経ソフトウェア, 4, pp.100-106 (2017).
- (3) 片山真一・中西通雄: “ソースコードと可視化グラフの連動動作を実現したソートアルゴリズム学習ツール”, 2009 PC カンファレンス論文集.
- (4) Steve McConnell: “Code Complete 第 2 版 下”, (株)クイープ 訳, 日経 BP 社 (2005).