

例題を利用するプログラミング支援環境

- 基本アルゴリズムの学習場面を例に -

土屋孝文*1・齋藤真琴・鹿内拓哉・原田翔一・松井浩紀

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

情報系基礎科目「アルゴリズムとデータ構造」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（解法）の理解、アルゴリズムに対応するコーディング（プログラムの生成）、プログラムの実行過程とアルゴリズムの対応（プログラムの解釈）の理解が相互に関係している。本研究は各領域に学習支援環境の設計と運用を検討している。支援環境の設計には以下の点を考慮している。

- ① 学習者の自然な自己説明や仮説生成を中心におく。
- ② 演習室における同時作業となるので、活動結果の集計/共有/比較や、学習者間コミュニケーションなど協調的な学習環境を活用する。
- ③ 講義は活動結果と理論的知識の橋渡しを行う。

アルゴリズム例題の理解については、①自分の（素朴な）解法の内省と、学習対象の（賢い）解法の具体的な操作事例に仮説的な説明を行う学習環境の運用、②自己説明や感想の共有環境の運用、③説明レベルや誤答の分析から、学習環境の再設計を行っている。基本的なアルゴリズムに関する運用事例では、事前に知識を準備せずとも、学習者は具体的な操作事例の観察を通して、背景の操作手続きルールを自然に模倣、推論できる²⁾。

一方、コーディングについては領域知識の適切な運用支援が必要である。昨年度は手続き的に重要な部分に穴のあるプログラムを完成させる課題に対し、適切な知識運用を促す段階的なヒント（関連知識プロンプト）を検討した（図1）。具体的には、穴周辺のプログラム量と穴埋め方法の切り替えが可能な環境である²⁾。図1は選択ソートの段階的ヒントページの例である。図1上のページから空所量が減少しながら、最後のヒントでは下のページのように解答方法が生成型から選択型となる。ここでは学習者にヒントとして与えられる解の一部を利用して、次の問題解決を進める活動が期待された。

しかし運用結果からは、41名のヒント利用者のほぼ全員が明示的にはプログラム生成を行わず選択型による回答へ進んでおり、より能動的に問題解決に関与しながら、関連知識の確認を自然に行える環境への再設計が課題となった。

本稿では、解の一部をヒントに残りの解の生成を続ける活動ではなく、解を構成する定型的なプログラムパターンや一般的な処理手続きの具体的な使用例（例題集）をヒントに、それらの適切な利用によって解の生成を促す学習環境の設計と運用を報告する。

以下ではソートを例にアルゴリズムの理解環境の改良を報告し、続いて例題集の構成と運用について述べる。

```
int main(){
    int i,j;
    int t; /* i番目の値はt */
    int k, min; /* 最小はk番目、値はmin */

    set_array();
    print_array();

    選択ソートで入れ替えてください

    print_array();

    return 0;
}

for (i = 0; i < N; i++) {
    k = i;
    解答群1
    min = a[i];

    for (j = i + 1; j < N; j++) {
        if (a[j] < min) {
            min = a[j];
            解答群1
        }
    }
    a[i] = a[k];
    解答群1
    ア a[i] = t;
    イ a[k] = t;
    ウ a[k] = a[i];
    エ t = a[k];
    print_array();
}
```

図1 選択ソート: 生成型ヒントと選択型ヒントページ例

2. アルゴリズム理解活動 - 基本ソートを例に

基本ソートの理解では、自分の解法を対象化する活動として、5つの数を並びかえてみるページを利用し、その操作履歴を参考にしながら自分の解法を説明、共有する。続いて学習対象ソートに関する動画を観察する。

今年度は対象のソート法に従った一連の操作を問うクイズページを改良した（図2）。初期系列が動的に生成され1手ごとに正誤フィードバックがなされる。学習者は正答までクイズを繰り返したあと、手

続きの説明と感想を共有する。なお、模倣することのできる手続きであっても適切な説明は難しく、支援が課題である。

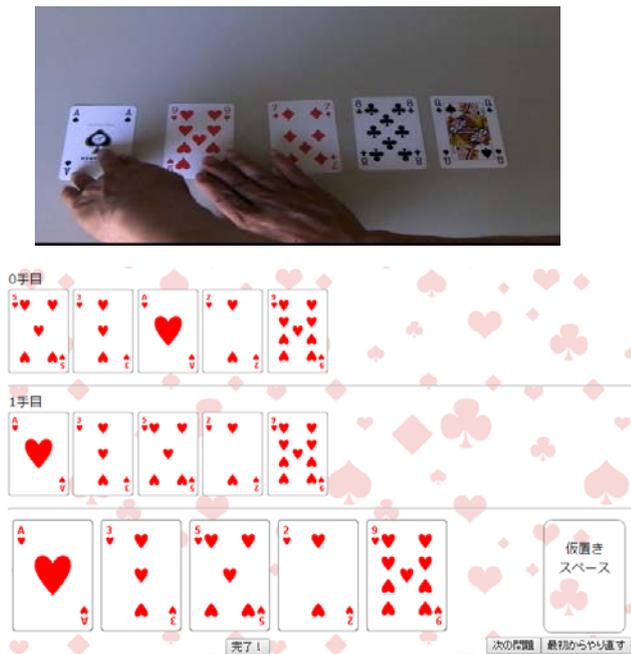


図2 観察・仮説生成用動画と仮説テストページ・選択ソート

3. 基本知識としての例題集 - 構成と運用

1980年代にはプログラミングの学習や専門的スキルの熟達を支える知識の構造（一般性）や性質（領域性）について、認知心理学における知識/記憶研究の視点から活発な議論がなされた。言語固有の構文知識や意味的知識（たとえば制御構造）のほか、類似の問題処理を一定程度抽象化した知識構造が注目された⁽¹⁾。プログラムを解釈する過程では、どのような問題解決法がとられているかを読み取るため、基本問題処理の知識（スキーマ知識）の想起と適用が重要とされ、同時にプログラムがどのように動くかの読み取りには、制御の流れと変数データの変化を特定するための知識（定型的なプログラムパターンなど）が必要との分析がなされた。

これらの知見からは、初学者に向けて基本的な問題処理知識に対応する小さく具体的な例題集を準備し、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知識の利用を促す学習環境が考えられる。学習者が例題の知識を十分内化できていない場合は例題集に戻って想起や再学習が可能な環境である。

例題には、保持可能な程度に適度な大きさであること、適当な量のスキーマ知識あるいは制御知識の具体例になっていること、学習者自身がいつでも例題コードに立ち戻って学習できること（学習者自身で一定の抽象的知識を得られる良質な例題であること）などの要件が求められるだろう。

学習対象のプログラムコードを分析し、9 カテゴリ（入出力、条件分岐、ループ、配列処理、文字列処理、二重ループ、関数、再帰、その他）について

14の例題を選択した。その他のカテゴリには、乱数生成や値の交換（swap）などが含まれる。図3は、二重ループの例題である。

PROG 6.2 1次元配列とループ

<問題>

0から19まで20文字を格納した配列を、以下のように出力せよ。

<実行結果>

```

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
6 7 8 9 10 11 12 13 14 15 16 17 18 19
7 8 9 10 11 12 13 14 15 16 17 18 19
8 9 10 11 12 13 14 15 16 17 18 19
9 10 11 12 13 14 15 16 17 18 19
10 11 12 13 14 15 16 17 18 19
11 12 13 14 15 16 17 18 19
12 13 14 15 16 17 18 19
13 14 15 16 17 18 19
14 15 16 17 18 19
15 16 17 18 19
16 17 18 19
17 18 19
18 19
19

```

プログラムを見る

図3 二重ループ処理の基本例題

たとえば、図1の選択ソート生成型ページのヒントに(A)二重ループ処理の例題と(B)値交換の例題の参照が追加された。この課題の運用結果(103名)では、一定数の学習者が課題解決時に例題コードに戻っている（ヒント(A)に16.5%、(B)に29.1%）。二重ループ処理は、この課題以前から繰り返し必要とされているため再参照数が減少しているが、値交換は、この課題が準備後に初出となったため比較して参照が多くなっている。例題集の準備と利用については参加者の97.8%より肯定的な評価がなされた。

4. おわりに

プログラムの解釈支援にも例題集の利用が考えられる。また例題集自身に実行過程に関する知識を追加できるだろう。今年度は、再帰処理の戻り値合成についてアキュムレータとの誤概念修正を目的に実行過程の図示を試みている。

図2の一連の操作結果は、プログラムトレースにおける変数変化の具体例にあたる。これらを利用して変数や処理構造を視覚的にアルゴリズムに対応づける試みを計画している。

参考文献

- (1) Detienne, F : "Expert programming knowledge: a schema-based approach", In T. R.G Green, R. Samurcay, & D. Gilmore (Eds): Psychology of Programming, Academic Press, pp.205-222 (1990).
- (2) 土屋孝文, 坂倉毅俊, 鹿内拓哉, 齋藤真琴, "段階的なヒントを利用するプログラミング -基本アルゴリズムの学習場面を例に-", PCカンファレンス論文集, pp.86-87 (2014).