

プログラミング教育における個人の理解度に応じた学習順序の決定

Tailor-Made Exercise Sequence for C Programming Education

田口 浩†
Hiroshi Taguchi

島川 博光†
Hiromitsu Shimakawa

1 はじめに

プログラミング教育において、初心者に対する教育は特に重要であり、慎重に行う必要がある。なぜならば、学習の初期に誤って理解した事項や十分に理解しきれなかった事項が放置されたまま学習が進められると、その後のプログラミング能力に大きく影響してしまうからである。また、学習初期に学習内容の理解に行き詰まってしまうと、学習に対する意欲を失い、挫折してしまう原因となる。

大学などにおけるプログラミング演習科目では、ほとんどの場合、すべての学習者に対して同じ順序で共通の演習課題を出題しているというのが現状である。しかし、プログラミングの学習を進めるにあたっては複数通りの学習順序が考えられ、その順序によって各学習者の理解のしやすさも変わる。よって、現状のようなスタイルで教育を行うと、定められた学習順序の適否や演習課題の難易度によって、学習者間で理解度に大きなばらつきが発生してしまうことは否めない。

これまでも、プログラミング教育を支援するための研究[1][2][3]は活発に行われているが、この問題点についてはほとんど未解決のままである。たとえば、文献[1]では学習者の理解度に応じて個人別に異なるプログラミング課題を出題する方法を提案するなど、それぞれの研究において、効果的なプログラミング教育を実現するための様々な方法が提案されているが、学習順序を学習者ごとに変更するといったことは行われていない。

この問題点を解決するためには、各々の学習者にとって最も学習効果が高まると思われる学習の進め方を判断し、それに沿って個別に異なる演習課題を出題することが望ましい。そこで我々は、C言語プログラミングの初心者を対象に、そのような効果的なプログラミング教育を実施することを目的とした、個別教育支援環境を提案した[4]。本稿では、その提案環境で実現している、各学習者の理解度に応じた学習順序の決定方法について、実際のプログラミング学習者に対して実施したアンケートの結果をもとに考察を行う。

2 C言語プログラミング教育支援環境

2.1 C言語の学習の進め方

文献[5]によれば、C言語の基礎的な学習内容は8つの単元に大きく分けられ、図1に示すように、最下位の「標準入出力」単元から最上位の「ファイル処理」単元および「メモリ管理」単元に向けて順に学習することが望まれる。最上位の「ファイル処理」単元と「メモリ管理」単元のように同位に位置する単元同士では、それらの学習順序は特に問われない。したがって、これらの学習内容をすべて学習するための学習順序は複数通り存在する。

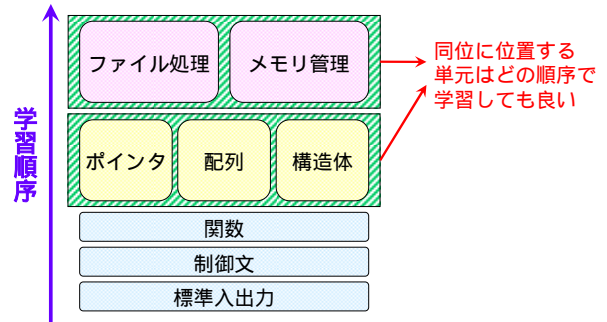


図1 学習内容の分類

2.2 リコメンデーションを利用した教育支援環境

我々は、C言語プログラミングの初心者を対象とした、個人の理解度に合わせて教育支援環境を提案した[4]。本環境は、各学習者の理解度や進捗を考慮して、個別の学習計画を自動的に策定するという特徴を有する。本環境では、学習者ごとの学習順序を決定するさいに協調フィルタリングによるリコメンデーション[6]を利用する。本節では、ある学習者に対してまだ出題されていない演習課題の成績を推測する場合を例に挙げて、この手法の概要を説明する。

リコメンデーションを実行するためにはまず、図2に示すような、行が学習者データ、列が演習課題データ、要素が成績である $p \times q$ 行列 X を作成する。ただし、その時点で未出題の演習課題の成績については非数 (NaN: Not a Number) として扱う。

次に、各学習者の特徴を表現するために、行列 X の行ごとに要素の平均値を算出し、各要素から該当する行の平均値を引いた行列 Y を作成する。ただしこのとき、NaN は計算の対象外とし、行列 X において NaN であった要素に対応する行列 Y の要素はすべて 0 とする。そして、この行列 Y を用いて次の2つの処理を行うことにより、まだ出題されていない演習課題の成績を推測する。

行列 Y の各行は、それぞれの学習者の特徴を表すべ

	演習課題1	演習課題2	演習課題3	...	演習課題q
学習者1	64	92	NaN	...	85
学習者2	93	77	84	...	96
学習者3	NaN	58	61	...	NaN
⋮	⋮	⋮	⋮	⋮	⋮
学習者p	70	NaN	94	...	63

図2 行列 X の一例

† 立命館大学大学院 理工学研究科

クトルであり、それら p 個の行ベクトルが構成する部分空間がすべての学習者の特徴を表す。特異値分解[7]を用いて、この部分空間をその特徴を特に表す p 個以下の行ベクトルで構成される空間で近似する。

ある学習者について、行列 X の行に相当するベクトルが与えられたとき、その各要素から要素の平均値を引いた行ベクトル t を作成する。その学習者が既に解いた演習課題の成績は変化しないものとする、 p 次の空間内で t が動き得る範囲は限定される。そして、未出題の演習課題の成績に制限はないものとしたうえで、 t が動き得る点のうち、で近似した空間に最も近い点を最小 2 乗法[7]を用いて求める。この求められた点を t が今後到達する点であるとみなし、その各要素に最初に与えられた行の要素の平均値を加えた値を、各演習課題の成績の推測値とする。

3 個人の理解度に応じた学習順序の決定

3.1 学習順序を決定するための材料

学習順序を決定するさいには、各演習課題について学習者が得た成績を用いる。各演習課題にはあらかじめ「while 文の制御式が正しく書けている」や「return 文を用いて適切な値を返している」といった評価観点で複数設定されており、教員はそれぞれの評価観点について「理解できている」、「理解できていない」、「どちらともいえない」のいずれかを選択することにより評価を行う。理解できているか判断しかねる場合や、その評価観点に記されている内容を用いずにプログラムを作成している場合に「どちらともいえない」を選択するものとする。これらの評価を点数化したものをその演習課題の成績とする。

3.2 リコメンデーションによる学習順序の決定

我々が提案した環境では、各単元の学習内容を、さらに複数の学習項目に分割して扱う。学習項目とは「1 行出力」や「関数へのポイント渡し」といった、これ以上分割することができない最小単位のプログラミングに関する技術的な事項である。各学習項目はそれに対応する演習課題を解くことによって学習するものとしている。各学習項目に対応する演習課題は数題ずつ用意されている。また、2.1 節で述べたように、学習の進め方はただ 1 通りしか存在しないわけではなく、複数通りの学習順序が考えられ、その順序によって各学習者の理解のしやすさも変わる。よって、ある演習課題を解き終えたとき、次に出題されるべき演習課題の候補が複数存在する場合がある。このような場合、それらの候補の中から最も学習効果が高まると思われる演習課題を選択して出題する必要がある。

そこで本環境では、三菱電機製の DIAPRISM リコメンデーションシステム[8]を利用して、2.2 節で説明した協調フィルタリングによるリコメンデーションを実行し、次に出題すべき演習課題を決定する。具体的には、図 3 に示すように、リコメンデーションの対象となる学習者に加え、同じ環境で学習した過去の学習者、および現在学習中の学習者についての、対象となる学習者がそれまでに学習したすべての演習課題と、次に出題すべき候補となっているすべての演習課題の成績から、リコメンデーションシステムは対象となる学習者の、候補となっている各演習課題の成績を推測する。本環境ではこの推測値に基づいて、次に出題するのに最も相応しいと思われる演習課題を決定する。

	学習者Aが 学習済みの 演習課題1	...	学習者Aが 学習済みの 演習課題p	次の候補と なっている 演習課題1	...	次の候補と なっている 演習課題q
対象となる 学習者A	83	...	78			
現在学習中 の学習者1	69	...	86	55
⋮	⋮	⋮	⋮	⋮	⋮	⋮
現在学習中 の学習者m	91	NaN
過去の 学習者1	NaN	NaN
⋮	⋮	⋮	⋮	⋮	⋮	⋮
過去の 学習者n	74	...	97	NaN	...	70

図 3 リコメンデーションの実行

4 学習順序の決定方法に対する考察

4.1 リコメンデーション実行結果の利用法

リコメンデーションの実行により得られた各演習課題の成績の推測値から、次に出題する演習課題を決定する場合、「最も推測値の高い演習課題を選択する」と「最も推測値の低い演習課題を選択する」の 2 通りの方法が考えられる。それぞれの方法には、以下に示すような特徴がある。

(1) 最も推測値の高い演習課題を選択する

長所：学習者が比較的理解しやすいであろう演習課題を出題することになるので、その演習課題で行き詰まる可能性は低く、学習者のやる気を失わせない

短所：向上心の高い学習者にとっては、簡単すぎて物足りなさを感じてしまうかもしれない

(2) 最も推測値の低い演習課題を選択する

長所：学習者が比較的理解しにくいであろう演習課題を出題することになるので、そのプログラムを完成させることにより、学習者の力を伸ばすことができる

短所：演習課題の内容を理解することができず、プログラムを完成させることが困難となり、学習者がやる気を失う可能性がある

4.2 学習者に対するアンケートの実施

上述の 2 つの方法はいずれも一長一短があり、一概にどちらのほうが効果的であると言うことはできない。そこで、これらの学習順序の決定方法に対する考察を行うため、プログラミングの学習者に対してアンケートを実施した。

アンケートは、2003 年 9 月から 2004 年 1 月にかけて立命館大学 理工学部 情報学科で開講された 1 回生配当科目「情報処理演習」のある 1 クラスの受講者 46 名を対象として実施した。この科目では図 1 に示した各単元の内容について、1 単元あたり 1 ~ 3 回の演習を行い、全 15 回の演習でそれらの学習内容を一通り学習する。今回はリコメンデーションは利用せず、すべての学生に対して同一の学習順序で、共通の演習課題を各回 6 ~ 10 題ずつ宿題として出題し、プログラムを作成させた。そして、次回の演習の開始時に各演習課題のプログラムを提出してもらったと同時に、アンケートに回答してもらった。アンケートは、各演習課題に対する学習者の自己評価を次の 5 つの選択肢の中から選択してもらった。

- a. 自信満々
- b. 大体理解できたと思う
- c. プログラムは自力で完成させたが、理解できたか不安
- d. 教員や友人から助言を受けながら何とか完成させた
- e. 何をすればいいのかわからなかった

4.3 アンケート結果に基づく考察

まず、C 言語プログラミングの初心者がどの単元で理解に苦しむのかをまとめたグラフを図 4 に示す。このグラフは、各単元に対応する演習課題のうち、5 分の 1 以上の演習課題に対して選択肢 c, d, e のいずれかを選択した学生の割合を示している。学生はこの科目で初めて C 言語プログラミングを学習するため、全体的に割合が高めになっているが、その中でも特に「ポイント」単元で理解に苦しんでいる学生が多いことがわかる。

学習順序を決定するにあたっては、学習者のやる気や意欲を低下させないことが重要となる。そこで次に、この「ポイント」単元に焦点を絞ってアンケート結果の分析を行う。「ポイント」単元のいずれかの演習課題に対して選択肢 e を選択した学生は 13 名であった。これらの学生が提出した当該課題のプログラムを見てみると、大きく次の 2 種類のパターンに分かれるということがわかった。

- ほぼ白紙同然で提出しているか、未提出である
- 未完成だったり、誤っていたりするものの、とりあえずプログラムを作成して提出している

前者のような学生の多くは、その後の演習課題に対して同様の傾向が見られた。これは「ポイント」単元の学習内容が理解できず、プログラムの作成に行き詰まってしまったため、プログラミング学習に対する意欲が失われたものと考えられる。よって、このような学生に対しては、極力行き詰まらないように、理解しやすいと思われる学習順序で学習させるほうが効果的であると推測される。一方、後者のような学生は、理解に苦しんではいないものの、学習に対するやる気は低下しておらず、その後の演習にも積極的に取り組み、理解しようと試みている場合がほとんどであった。したがって、このような学生に対しては、理解しにくいと思われる学習順序を選択したほうが、より能力を伸ばすことができ、効果的であると推測される。

このように、各学習者にとって最も学習効果が高まる学習順序を決定するさいには、学習者の理解度に加え、学習への取り組み姿勢も考慮したうえで、上述の 2 つの決定方法のいずれかを選択するべきである、ということがアンケート結果より明らかになった。

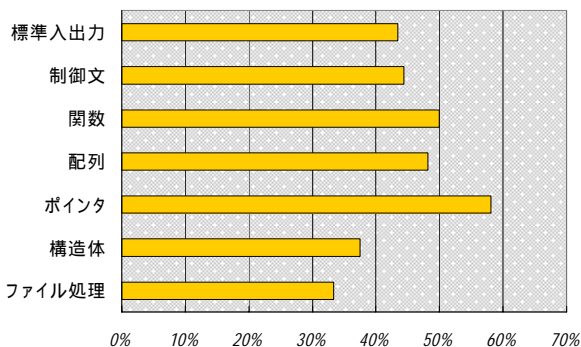


図 4 アンケート結果

4.4 教育支援環境への反映

我々が提案した教育支援環境では、学習者ごとの詳細な学習履歴を学習カルテと呼ぶ電子的な情報として保存しており、各機能を実行するさいにはこの学習カルテから必要な情報が自動的に検索される。今回のアンケート結果より導かれた学習順序の決定方法を本環境に実装するには、プログラムなどから教員が評価した、学習者の取り組み姿勢を学習カルテに追加すればよい。そして、その取り組み姿勢とリコメンデーションの結果から、次に出題するのに最も相応しいと思われる演習課題を判断することによって、より適切な学習順序で学習させることが可能となる。

5 おわりに

本稿では、C 言語プログラミングの初心者を対象としたプログラミング教育を効果的に実施するために、各学習者の理解度に応じた学習順序の決定方法について述べた。さらに、実際の学習者に対してアンケートを実施し、その結果に基づいて、その方法の考察を行った。

今後は、今回導かれた方法を実現できるように教育支援環境を改良し、その有効性の検証を行っていきたい。

参考文献

- [1] 竹田 尚彦, 川口 清志, 浅見 美紀, 佐合 尚子, “プログラミングの個別指導のための演習問題サーバ”, 情報処理学会研究報告 コンピュータと教育 No.58-4, pp.21-28 (2000).
- [2] 高本 明美, 藤井 美和子, 泉 直利, 田中 稔, “誤り原因を指摘するプログラム学習支援システムとその学習効果”, 教育システム情報学会誌 Vol.17 No.4, pp.533-540 (2000).
- [3] 諏訪 正則, 倉澤 邦美, 鈴木 恵介, 森本 康彦, 横山 節雄, 佐々木 整, 宮寺 庸造, “プログラミング教育における学習履歴取得システムの開発”, 第 2 回情報科学技術フォーラム(FIT2003)講演論文集, pp.583-584 (2003).
- [4] 田口 浩, 島川 博光, 石井 篤, “個人の理解度に合わせた C 言語プログラミング教育支援環境”, *Proc. of the 2nd International Conference on Creating, Connecting and Collaborating through Computing*, 日本語セッション, pp.60-63 (2004).
- [5] B.W.Kernighan, D.M.Ritchie, “The C Programming Language, Second Edition”, Prentice Hall (1988), 石田 晴久 訳, “プログラミング言語 C 第 2 版”, 共立出版 (1989).
- [6] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, John Riedl, “GroupsLens: An Open Architecture for Collaborative Filtering of Netnews”, *Proc. of the 1994 ACM Conference on Computer Supported Cooperative Work*, pp.175-186 (1994).
- [7] Gilbert Strang, “Linear Algebra and its Applications”, Academic Press (1976), 山口 昌哉 監訳, 井上 昭 訳, “線形代数とその応用”, 産業図書 (1978).
- [8] DIAPRISM リコメンデーションシステム 詳細情報, <http://www2.mdit.co.jp/service/diaprism/seihin/tjirei6-1.pdf>