

トレース表を用いたプログラミング学習者の理解過程分析手法[†]

尾崎 弘幸*

An Analysis Method of Programming Learners' Comprehension Process Using Trace Table

Hiroyuki Ozaki

1 はじめに

今日の大学におけるプログラミングの授業は講義と演習によって構成されている。プログラミングへの理解を高めるためには、アウトプットである演習はプログラミング学習において重要である。一方で、プログラミング演習において手が止まってしまうなどの困難を感じている学習者が見られる。そこで、教授者が学習者の演習活動をオンラインエディタなどを用いてモニターし、学習者の躓きなどを把握することで、支援を必要としている学習者を早期に発見しサポートする方法などが提案されている[1]。しかしながら、これらのツールは学習者がコードを1行も記入していないなど、演習問題に対して停滞している学習者に対して支援を行うことが難しい。また、そのような学習者に対して教科書の例題のプログラムの動きやその意味について深く問うと、あやふやであることがある。このことから、プログラミングが苦手な学習者は、プログラムを正しく読解できていない可能性がある。

本研究では、学習者の作成したトレース表から、学習者がどのようにプログラムを理解したのかを分析する。また、トレース表の作成が学習者のプログラムに対する理解を向上させる可能性について考察を行う。これは、演習時に停滞してしまうような学習者に対しても、プログラムの理解を促すことでプログラミング学習者の停滞を防ぐことが可能であると考えられる。

2 提案手法

本研究では学習者のプログラムの読解に着目する。プログラミングを学習する際、始めに学習者は教科書の例題からプログラムの動きや処理の流れなどを学ぶ。その後、例題に対応した演習問題を解くというのが一般的な流れであ

る。演習問題は例題の一部を改変したり、例題に類似する要素を用いて解くことが可能のように構成されている。しかし、プログラミングが苦手な学習者はその段階で困難を感じていることが多く、学習者の中には教科書の例題プログラムを理解しようとするのではなく、丸暗記しようとするケースもある。著者らは、このようなことが生じる原因の1つとして、例題プログラムの理解の不足や誤りがあると考えた。

プログラミングにおけるデバッグ支援で用いられているものにトレース表がある。これは、図1に示すようにプログラムの処理の流れに合わせて各ステップごとの変数の値などを記入していくものである。そのため、正しいトレース表を作成するためには、学習者はプログラムの各行がどの順番で実行され、各行の処理で変数の値がどのように変化するかを正確に読み解かなければならない。一方で、このようなトレース表は「プログラムの記述と比較して作成が容易である」という特徴がある。

もし学習者のプログラムに対する理解が誤っている場合、学習者が作成したトレース表にはその誤りの影響が表れると考える。本研究では、学習者がトレース表を正しく作成することができる状態がプログラムを正しく読めている状態とし、学習者がトレース表をどのような流れで作成し、学習者の理解がどのように変化したのかを分析する。また、学習者がトレース表を作成することによってプログラムを

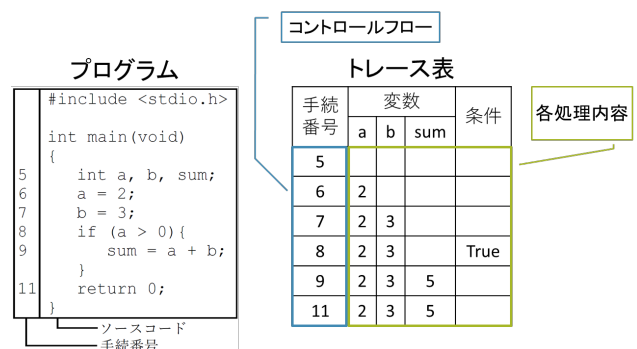


図1 本研究で用いるトレース表

[†]本研究の一部は以下において発表した
・日本ソフトウェア科学会第8回実践的IT教育シンポジウム (rePit2022)

*電子情報メディア工学専攻 2208005 橋浦研究室

正しく読めるようになることを目指し、目標達成のためにトレース表作成のための環境を作成する。なお、本手法で扱うプログラムについては、初学者のプログラムに対する理解の向上から、教科書等で例題などに用いられる比較的シンプルなものを対象とする。

3 ツール

2章で述べた学習者のプログラムに対する理解を向上させる方法として、トレース表の作成から採点までを自動的に行うことが出来る環境を実現する。また本研究では、学習者のトレース表作成過程を分析するため、学習者がトレース表を完成させるまでに教授者は何度もトレース表の採点を行わなければならない。加えて、採点の履歴を学習者ごとに正確に記録することは教授者の大きな負担となるため、自動化によりそのような問題を回避する。

本学では学習者は自身のノートパソコンを用いて演習を行うため、使用している OS が異なるなどの事態が考えられる。そのためツールの実現にあたってはインターネット接続とブラウザのみで利用可能なウェブアプリケーションとした。これにより、学習者間で異なる環境のコンピュータに合わせたツールをインストールすることなく、ツールの使用が可能になる。

学習者はブラウザからウェブアプリケーションの URL にアクセスし、画面のプルダウンメニューからトレース表の作成をする問題を選択する。トレース表の作成画面は画面左半分がプログラムのソースコード表示部、右半分がトレース表の入力部となっている。なお、本研究では初学者のプログラム読解に着目するため、学習者によるプログラムの編集は行わず、学習者は表示されたプログラムのトレースのみを行う。学習者は表示されたプログラムのソースコードを読み、トレース表の入力部に値を入力する。全ての入力が完了したら画面下の採点ボタンを押すことで、即

時に採点が行われ、誤りがある場合は誤っている箇所が赤くハイライトされる(図 2)。学習者は赤く表示された箇所を修正し、誤りが全てなくなるまで修正と採点を素早く何度でも繰り返すことができる。学習者が作成したトレース表の情報は、採点ボタンを押すごとに DB に自動的に格納される。

4 実験

本研究では学習者がどのような流れでトレース表を作成し、どのような過程でプログラムを理解するのかを調査するため 3 章で作成したツールを用いて実験を行った。

4.1 対象と実験の流れ

実験は本学先進工学部情報メディア工学科の学習者 22 名を被験者とした。本実験は「プログラミング I」が全て終了したタイミングで行い、実験の内容は講義で履修済みの範囲とした。本実験は、まず始めにプログラミングクイズを実施し、その後 3 章で作成したツールを用いてトレース表の作成を行うという流れで実施した。

4.2 プログラミングクイズ

プログラミングに関する知識を測る方法として、一般的にはペーパーテストが用いられている。本実験では学習者のプログラミングに関する知識を測る方法として多肢選択による問題を作成し、それをプログラミングクイズとして出題した。プログラミングクイズ作問にあたっては、プログラミング講義で使用している教科書[2]を用い、学習者がすでに履修済みの内容のみを出題した。また、問題の形式は 4 択とし、全 50 問(各 2 点、100 点満点)とした。具体的な実施は Microsoft Forms を用いた。

4.3 トレース表作成問題

学習者に、トレース表の作成を行った経験がないことも考えられるため、トレース表作成に先立ってトレース表の作成方法の簡単な説明と練習を行い、その後トレース表作成問題を実施した。

実験に用いたトレース表作成は 2 問で、トレースするプログラムは教科書の例題プログラムを用いた。初見のプログラムではなく授業中に出てきたプログラムを用いた理由は、2 章で述べたように既に履修済みの内容であってもプログラムを正しく読むことが出来ない学習者が一定数存在することが予想されたためである。

5 結果と考察

トレース表作成問題では 22 名中 19 名(約 86%)の学習者が最終的にトレース表を完成させることが出来た。また、ほとんどの学習者が 1 度の採点でトレース表を完成させる

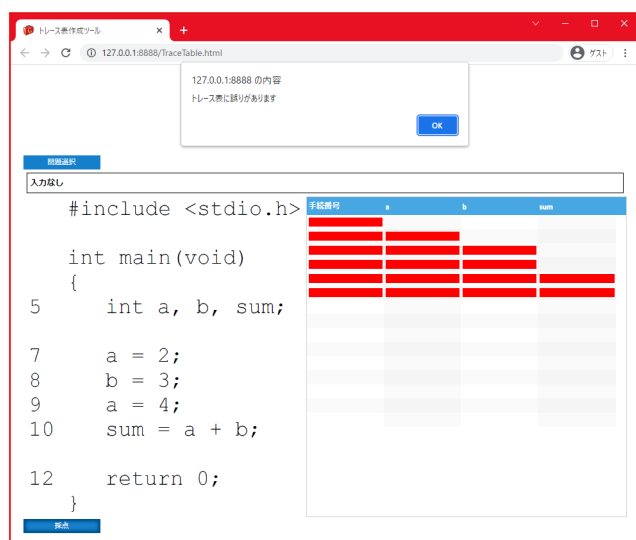


図 2 トレース表未記入で採点した際の画面

のではなく、ツールによるフィードバックを基に、複数回に渡ってトレース表の修正を行い、最終的にはトレース表を完成させている。よって、本ツールのフィードバックが学習者のプログラムの理解向上に寄与していると考えられる。

次に、トレース表を作成することで学習者がどのようにプログラムを理解したのかを考察するために、トレース表の作成開始から終了までの過程を Grounded Theory Approach(GTA)[3]を用いて評価を行った。本研究では 4.3 節で記録された学習者のトレース表のデータからプログラムを GTA の入力として扱う。

5.1 GTA を用いた分析の流れ

データの考察は GTA に基づいて、次の手順で行う。

1. トレース表の切片化
2. テキスト表作成
3. カテゴリー関連図
4. カテゴリー関連統合図

まず始めにツールが記録した学習者 1 人分のトレース表を切片化しテキスト表を作成する。その後、テキストの表をカテゴリー関連図にすることで学習者 1 人分のトレース表作成の流れを掴む。この作業を学習者 22 名分行い、最後

に作成された 22 個のカテゴリー関連図を 1 つに統合することで、最終的なカテゴリー関連統合図を作成する。

5.2 トレース表からカテゴリー関連統合図の作成

カテゴリー関連図を作成するにあたり、まず初めに学習者がトレース表を完成させるまでの過程を切片化する必要がある。そこで本研究では学習者の作成したトレース表 1 枚 1 枚を 1 つのシーンとした。その後、切片化したシーンについてプロパティとディメンションを用いてテキスト表を作成しラベルを付与する。これにより、1 枚のトレース表における学習者が行った行動がラベルとしてあらわされる。次にラベルを近いもの同士でまとめカテゴリー化し、それらを時系列順に矢印で関連付けたものがカテゴリー関連図となる。よって、カテゴリー関連図は 1 人の学習者が行ったトレース表の作成過程を表したものになる。最後に、個別に作成したカテゴリー関連図を 1 つに統合したものがカテゴリー関連統合図となる。なお個別のカテゴリー関連図を 1 つに統合する際に重複する関連は数値と太さで表している。

5.3 トレース表作成過程

図 3 に問題 1 から作成したカテゴリー関連統合図を示す。

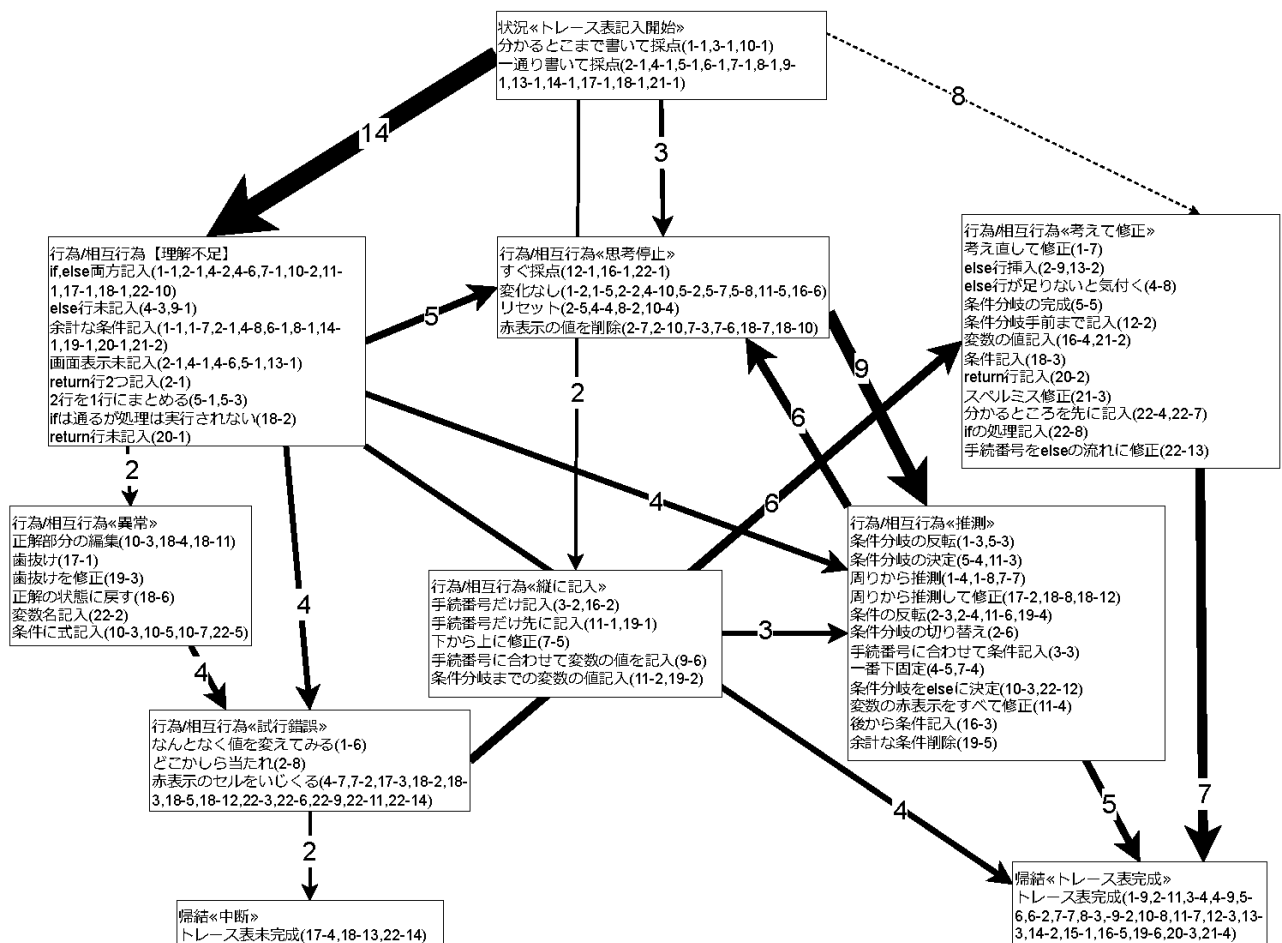


図 3 問題 1 のカテゴリー関連統合図

カテゴリー関連統合図を作成した結果、カテゴリー間の関連を示す矢印がトレース表の作成開始からその終了までの間を複雑に行き来していることが明らかになった。このことから、ほとんどの学習者がトレース表を書き始めてから誤りを犯すことなくストレートにトレース表を完成させるのではなく、一度トレース表の作成が停滞し、いくつかの試行錯誤をする中でプログラムを理解し、最終的にトレース表を完成させていることが明らかになった。

5.4 特徴的なトレース表作成

通常、学習者はプログラムの実行順に合わせてトレース表の作成を行う。しかし、一部の学習者ではそのような解答方法を取らない特徴的なものが見られた。それら特徴の1つにトレース表の手続番号を先に記入しその後、変数の値などを記入していくものがある(図4)。これは、本ツールのフィードバックから、プログラムの実行順序を推測し、それに合わせてトレース表の作成を行っているものと考えられる。そのため、トレース表作成過程においてこのような特徴が見られる学習者は特にプログラムの読解に困難を感じていることが明らかである。

図4 手続番号を先に記入するトレース表の例

5.5 プログラミングクイズとの関連

図3のカテゴリー関連統合図が示すように、多くの学習者はトレース表の完成までに試行錯誤を繰り返す。そのため、その行為が多い学習者ほどトレース表の修正回数が増えることになり、そのような学習者はプログラムの読解に困難を感じていると言える。しかし、本実験で実施したプログラミングクイズの正答率とトレース表修正回数の間には相関が見られない結果となった。これは、ペーパーテストのような従来の方法ではプログラムの読解に困難を感じている学習者の発見が難しいことを表していると考えられる。

6 関連研究

Risha[4]らはプログラムの最終的な出力を答えるトレース学習よりも、プログラムをステップごとにトレースすることが初学者にとっては重要であるとして、学習者がトレ

ース表を作成できるWebアプリケーションを開発している。彼らは研究の中で、ツールの機能として自動で行われるトレース表の作成過程を学習者が確認できるものを実装した結果、多くの学習者が自身でトレース表の作成を行わずに答えを確認することを目的にその機能を使用したとしている。本研究では、学習者はトレース表の作成を一から全て行うことが求められ、さらに学習者に対するフィードバックも誤りの箇所をハイライトすることに留めている。そのため、学習者は能動的にプログラムを読みその動きを理解することが求められるため、プログラムへの理解がより促進されることが考えられる。

7 結論と展望

本研究では、学習者がトレース表を完成させることでプログラムの理解が向上している可能性を示し、それを支援するためのトレース表作成ツールを開発した。また開発したツールを用いて学習者のトレース表作成過程を記録し、それらを分析することで学習者がどのようにしてトレース表の作成を行っているのかを調べた。その結果、学習者はトレース表の採点から自身のプログラムに対する理解の誤りに気付くことで、その誤りを修正しトレース表の作成を行っていることが判明した。このことから、本研究で提案したトレース表作成が学習者のプログラムの理解向上に寄与していることが示唆された。

今後の課題としては、学習者の理解が誤っている箇所をより詳細に分析することで、学習者が陥りやすい誤解をあらかじめ把握しその兆候が見られる学習者に対して積極的に支援を行うことなどが挙げられる。

参考文献

- 井垣宏, 齊藤俊, 井上亮文, 中村亮太, 楠本真二, “プログラミング演習における進捗状況把握のためのコーディング過程可視化システム c3pv の提案”, 情報処理学会論文誌, vol.54, no.1, pp.330-339, Jan. 2013.
- 柴田望洋, 新・明解 C 言語入門編, 野沢喜美男, Ed. SBクリエイティブ株式会社, 2017.
- 戈木クレイグ・ヒル滋子, グラウンデッド・セオリー・アプローチを用いたデータ収集法, ser.グラウンデッドセオリー・アプローチを用いたデータ収集法 新曜社, 2014.
- Z. Risha, J. Barria-Pineda, K. Akhuseyinoglu, and P. Brusilovsky, “Stepwise help and scaffolding for java code tracing problems with an interactive trace table,” 21st Koli Calling International Conference on Computing Education Research, no.27, pp.1-10, Nov. 2021.

指導教授	審査委員 (主査)	准教授	橋浦 弘明
	審査委員 (副査)	教授	糸野 文洋
	審査委員 (副査)	准教授	松田 洋
