

## トレース課題における誤りの段階的学習支援 —生成 AI を活用した誤り箇所特定を応用して—

### Step-by-step notice for errors of semantics in tracing tasks -Error detection in coding task using generative AI-

竹内 寛典<sup>\*1</sup>, 松浦 健二<sup>\*2</sup>

Hironori TAKEUCHI<sup>\*1</sup>, Kenji MATSUURA<sup>\*2</sup>

<sup>\*1,2</sup> 徳島大学情報センター

<sup>\*1</sup>Center for Administration of Information Technology, Tokushima University

Email: takeuchi.hironori@tokushima-u.ac.jp

**あらまし** : プログラミングの初学者が課題に取り組む際に, RAD/IDE 環境下においても Syntax としてはエラーではない意味的な誤りには気づきにくいところがある. 生成 AI を利用した場合, Syntax は修正されることがあるが, インタクションの工夫によって意味的な誤りも指摘できる可能性がある. そこで, 本研究ではトレース課題を取り上げて, 生成 AI を応用することで誤り箇所を特定し, それを段階的に提示する学習環境を設計している. 本稿では, その一次評価について述べる.

**キーワード** : プログラミング, トレース問題, 生成 AI, ヒント

#### 1. はじめに

これまでの高等教育機関以降での対象とされてきたプログラミングの教育・学習は, 初等教育課程から開始されるようになり, かつ社会に出てからも必須となっている. プログラミング的思考は, 対象の変化を論理的に捉えることができるようになり, 現代においては特定のシステム開発能力のみならず, 初学者からその継続的スキル開発も含む生涯学習の対象と言える.

こうした中, デバッグやコード補間を伴う開発環境は次々とアップデートがなされている. また, そのような環境に応じて, 様々なプログラミングスキル開発のための学習支援環境も学術的提案がなされている. 例えば, コード補間やデバッグ機構を備えた環境では, コンパイラ等がエラーを含むコードの問題点を特定することができるが, それを応用する学習支援システムが提案されている<sup>(1)</sup>. また, プログラムの振る舞いトレースからの理解を促す学習支援環境も提案されている. 後者は, ソースを書く学習に対して読む学習に着目した研究といえ, 本研究でも特に参考になった研究である. なお, プログラミングにおけるトレース問題については, 文献<sup>(2)</sup>を参考にした.

このような研究が多数展開されてきた中で, ChatGPT をはじめとした生成 AI では, 例えば ChatGPT の Codex を用いることで, 所望のコードの自動生成や翻訳といった様々な利点を有する機能が提供されている. そこで, コードを書くというモチベーションにせよ, コードを読むというモチベーションにせよ, こうした生成 AI の機能をプログラミング学習支援に取り込むのは今や自然な着想といえる. 特に, インタクション設計を通じて, Syntax レベルではなく Semantic レベルでの任意の題材を対象とした学習支援が可能と考えられる. そこで本研

究では, 初学者を対象に, プログラムトレース課題を採り上げ, 生成 AI を用いた学習支援環境の開発を目的に試作した<sup>(3)</sup>. 本稿では, その概要と一次評価報告を行う.

#### 2. 目的に応じたコード化の学習環境設計

##### 2.1 誤りへの気づき

プログラミング学習の初学者の陥り易い誤りは様々な研究で報告されている<sup>(4)</sup>. 誤りの分類も書く研究でなされているが, コンパイル時のエラーとしては表示される場合の多くは, 開発環境自体が要修正箇所の指摘が行えるということである. 一方で, デバッガやコンパイラが抽出できない意味的な誤りについては開発環境で取り出すことが困難である. 文献<sup>(5)</sup>を参考に, 本研究ではこのような誤りやすいステートメント群として以下の4種を採り上げる.

- A) 3つ以上の条件を有する if 文
- B) if 文のネスト構造化
- C) 多重の for 文
- D) 配列を入れ替える際の index 操作

##### 2.2 学習におけるコード修正過程

本研究では, 文献<sup>(3)</sup>におけるデバッグ過程のモデル化を応用し, 以下の4過程のスパイラルとして設計している. なお, 本稿での4過程では, システム要件から, 初期には学習者自身で設計・実装を行って, その挙動を見ていく2サイクル目から記載していることに注意されたい.

- (1) 修正仮説策定(誤り原因の推測)
- (2) 修正プラン生成(推測された誤りの修正箇所を特定し, 全体からの修正方針を策定)
- (3) 修正実行(プランに基づき, 実際のコードを修正する)
- (4) 動作評価(修正されたコードを実行し, 想定さ

れたかどうかを評価)

ここで、本研究における具体的な学習対象課題はプログラムトレース課題として簡易化している。トレース課題の場合、対象の特定が局所的なところに集約されることが多いためである。

## 2.3 ヒント

トレース課題における意味的誤りは、例えば変数値をデバッガで調査できる場合もあるが、先に述べた典型的な誤りにデバッガで全て対応できる訳ではない。また、初学者は思い込み等によって、自らの誤り箇所気づかないことも多いため、段階的なヒントを設計している。このヒント生成は、バックエンドに生成 AI(評価時点では ChatGPT3.5Turbo のモデルに対する API<sup>4)</sup>)を利用して任意のコードに対応可能と考えているが、検証を重ねたところ、指示の方法や学習者の入力コードによって、必ずしも適切な応答とは言えず、そのため、ある程度のテンプレート的なコードを用意して抑制している。ヒント機能では、正解表示も含めて3レベル設定している。

レベル1は誤り推定箇所のハイライト表示であり、行単位での見直しを促している。レベル2は、選択表示であり、1行ごとにマスク処理を行って、学習者が事前に検討できた行から選択して開いていく操作を伴うこととしている。このようなレベル訳により、気づきのフェーズから、自身で考え確認するフェーズへと移行できる。

## 2.4 学習環境

本システムは、フロントエンドにはブラウザによるアクセスとして React を用いた環境を提供し、問題提供やその修正環境とした。サーバ側のシステムは問題提示と生成 AI とのバックエンドインタラクションを担い、学習記録保持や誤り特定、生成 AI の応答加工処理、学習者のコード実行環境、採点機構などを有している。

クライアントの画面では、簡易パネル上で問題や制約条件が提示され、コード編集パネルでコードを編集し、実行ボタンやヒントボタンによってサーバとの非同期インタラクションを実現した。

## 3. 実験的試用

本試作システムの実験的評価のための試用を行った。被験者は大学生を中心とした初学者 20 名であり、対象言語には Python とした。被験者は情報系の学生であり、基本的な Python のコードは記述できる。本研究は Syntax レベルを対象としないため、Web でのそのレベルに対する参照は許容している。A 群を提案機能利用群として、B 群は提案機能としてのヒント利用のない群とした。学習時にはそれぞれ同じ問題を 2.1 節の種別に応じて各種 1 問で 4 問構成し、前後に事前・事後のテストを行った。なお、実験参加エントリが A 群 10 人に到達した時点で、以降 B 群で 10 人集まるまでという形で振り分けを行ったた

め、初期状態は考慮していない。

対象はトレース課題であるため、前後のテストや学習の対象はそれに合致した内容として準備したが、標準入力などのタスクと直接関係しないところはテンプレートで用意した(任意のコードに対応させる際にはこのような省力化も見直す予定である)。

テストの平均点を前後比較行ったところ、A 群では 19.21 点の向上、B 群では 10.84 点の向上が見られ、得点向上においては A 群がより高い効果となった。ただし、初期状態を考慮せずの群分けであったことから、B 群の初期の方が約 9 点高かったため、伸びではなくて事後テスト比較をしてみたところ、ほぼ同程度の点数になっていた。このことから、継続的な検証を改めて行うことが望まれる。

## 4. おわりに

本研究では、生成 AI を応用してプログラムトレース課題へのヒント提示機構を実現した。本環境を用いた有用性が明らかになったとは明言できないが、点数向上ということでは一つの示唆を与えている。問題の数や種別を増やし、被験者を増やした適切な検証を改めて行っていきたいと考えている。また、試用のために、問題に対応する制約を一部設けていたが、任意の課題に対応できるようにシステム改良を重ねたいと考えている。

### 謝辞

本研究実施に際しては、システム開発において矢代涼氏に助力いただいた。ここに謝意を表す。

### 参考文献

- (1) 近藤亮太, 名倉正剛: “コンパイルエラーメッセージの分類に基づく初学者へのプログラミング学習支援手法”, 実践的 IT 教育シンポジウム rePiT 論文集, 一般社団法人日本ソフトウェア科学会, 実践的 IT 教育研究会, p. 63-74 (2023)
- (2) 松爲泰生, 前田新太郎, 茂木誠拓, 古池謙人, 東本崇仁: “振舞いのトレースを通してソースコードの機能の理解を促す学習支援システムの開発”, 先進的学習科学と工学研究会 94 回, 人工知能学会, p. 13-18 (2022)
- (3) 江木鶴子, 竹内章: “プログラミング初心者にはトレースを指導するデバッグ支援システムの開発と評価”, 日本教育工学会論文誌, Vol. 32, No. 4, pp. 369-381, (2009)
- (4) OpenAI, ChatGPT, <https://chat.openai.com> (閲覧日: 2024 年 2 月 3 日)
- (5) 矢代涼, 松浦健二, 竹内寛典: “生成 AI を活用したデバッグ機構を通じて誤りに気づかせるコーディング学習支援環境”, 教育システム情報学会 2023 年度学生研究発表会(四国地区), pp. 165-166 (2024)
- (6) Neil CC, Brown and Amjad Altadmri: “Novice java programming mistakes: Largescale data vs. educator beliefs”, ACM Transactions on Computing Education (TOCE), Vol. 17, No. 2, pp. 1-21 (2017)