

プログラミング学習における作問学習の自動評価の適用可能性について Feasibility of Automatic Evaluation in Programming Problem-Posing Learning

山本 明澄¹⁾ 蜂巢 吉成¹⁾ 吉田 敦¹⁾ 桑原 寛明¹⁾
Akito Yamamoto Yoshinari Hachisu Atsushi Yoshida Hiroaki Kuwabara

1 はじめに

プログラミングの学習方法として、作問学習が挙げられる。作問学習とは、学習者自身が問題を作成するような学習方法である。作問学習は出題された問題文に対して解答する記述問題と比べて、問題やその解法に対する深い理解が必要なため、より高い学習効果が期待できる。しかし、作問学習には、評価が難しいという問題がある。評価が難しい原因は、評価基準が曖昧なことである。評価基準は出題者が定義する必要があり、学習者の提出物に対してそれぞれに適した評価が必要なので、手間がかかる。

本研究では、プログラムの作問学習の評価を生成 AI を用いて行う方法を考察し、学生らが作成した問題の評価結果について述べる。

2 関連研究

佐藤 [1] は、プログラムを作らせる問題文を学習者自身に作らせる作問学習について、評価基準を明確化し、作問学習の定量的評価を行った。佐藤が提案したルーブリックを表 1 に示す。表 1 は文献 [1] の p.2 表 1 を筆者が書き直したものである。

文献 [1] では作問課題の出題例として、3 分岐以上の if 文の構造に関する問題が取り上げられている。作らせるプログラムの構文制約として、if を使わせること、3 分岐以上の条件分岐を行わせること、prompt 関数を用いて変数への入力を行わせること、結果の表示に連結演算子を用いること、が示されている。

ルーブリックで高評価となる問題の例として、クリスマスパーティのために、予算 4000 円で 100 円ショップで購入した品物の個数を入力すると、合計金額に応じて「〇〇円は買いすぎだね」「〇〇円は予算内だね。いいパーティが出来そう」「〇〇円ならもう少し買えそうだね」「〇〇円は少なすぎるかな」などのメッセージを表示する問題が挙げられている。数値の入力に加えて算術演算を必要とすることなどが高評価につながっている。

低評価となる問題の例として、バスケの試合の得点を入力すると得点に応じて「もっと頑張ろう」「その調子」「すごいね」と表示する問題が挙げられている。表示に連結演算子を用いていないことや、練習問題の数値を変えただけという点で低評価になっている。

学習者を作問課題の取り組みの有無で二群に分けると、作問取組群が作問非取組群よりも小テストや練習問題に対する平均解答所用時間が短く平均正答率が高いことを示した。また、作問課題の評価点が高い学習者は解答所用時間が短く平均正答率が高いことも示した。なお、作問課題の評価はルーブリックに基づいて人手で行っている。

大石ら [2] は高校の「情報 I」のプログラミング単位に関する作問学習で作成された問題を対象として、対話型生成 AI による診断・評価、フィードバックの結果を

表 1 佐藤のルーブリック (文献 [1] p.2 表 1 を元に筆者が作成した)

評価の観点	評価の尺度		
	3 (十分に満たしている)	2 (一部は満たしている)	1 (満たしていない)
作成された問題は単元の理解を深めるものか	2の尺度を満たし、かつ単元理解の強いに沿った追加の要素を含んだ問題文である	作問課題の条件を満たした問題文である	作問課題の条件を満たしていない問題文である
オリジナリティ・創意工夫はあるか	2の尺度を満たし、かつ独自の工夫がある	講義で取り組んだ練習問題とは異なった問題文である	講義で取り組んだ練習問題と同じ、もしくは少し変えただけの問題文である
問題文の表現として必要な情報提供ができていないか	2と3の尺度のいずれでもない	解答者のプログラム作成が容易になり過ぎる過剰な情報を提示している	解答者がプログラムを作成するのに必要な情報が不足している
作成した問題に対しての解答例は適切か	2の尺度を満たし、かつ作成された問題から作られる最も無駄のないプログラムである	作問課題の条件を満たした解答例である	2の尺度を満たさない、もしくはプログラムとして間違っている

分析している。その結果、対話型生成 AI は、適切にプロンプトを設定すれば、学習者が作成した問題を診断・評価でき、学習者が設定した目標に沿ってフィードバックできることを示した。

本研究では、プログラムを作らせる作問学習において、佐藤が提案したルーブリックに従った評価を生成 AI を用いて行う方法について検討した。

3 生成 AI を用いた評価方法の提案

3.1 概略

本章では、プログラム作問学習における生成 AI を用いた評価方法を提案する。本研究における作問学習・評価は次の 3 つのステップで行われる。流れを図 1 に示す。

1. 出題者はプログラミング作成問題を出題する
2. 学習者は出題に対して解答する
3. 生成 AI により解答の評価を行う

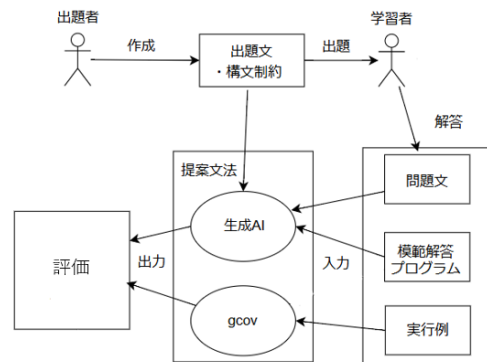


図 1 作問学習における出題、解答、評価の流れ

出題者はプログラムを作成するような問題の出題を行う。制約としてプログラムで使用する構文要素を指定する。出題文の例を図 2 に示す。

学習者は出題に対しての解答として、出題文に対する問題文、問題文から作成される模範解答プログラム、模

1) 南山大学

以下の構文制約を満たす C 言語プログラムを作成するような問題文, その模範解答プログラムと動作確認に必要な実行例を解答してください。

構文制約

1. while(1) または for(;;) を使用して無限ループを作成すること
2. ループ内で scanf 関数を使って整数を入力させ, その数値を使用して何らかの計算や処理を行うこと
3. ある条件を満たすときに if 文でチェックし, break を使ってループを終了させること
4. ループ終了後に計算結果や処理結果を表示すること

図 2 無限ループの出題文

表 2 提出物の評価基準と評価方法

提出物	評価基準	評価方法
問題文	表 1 の 1, 2, 3 行目	生成 AI
模範解答プログラム	構文制約を満たしているか	生成 AI
実行例	ステートメントカバレッジ	gcov コマンド

範解答プログラムの実行例をそれぞれ作成し提出する。学習者が解答した問題文, 模範解答プログラム, 実行例に対して, 評価をそれぞれ行う。本研究では, 問題文と模範解答プログラムの評価に生成 AI を使用する。なお, 生成 AI には ChatGPT-4o を利用する。

3.2 評価基準

評価基準としては, 問題文に関しては, 佐藤 [1] のループリックを用いる。「作成された問題は単元の理解を深めるものか」(以降, 単元理解) は出題文に提示した構文制約を満たしているか, 「オリジナリティ・創意工夫はあるか」(以降, 創意工夫) は作成者自身が考えたことが分かるような問題文か, 「問題文の表現として必要な情報提供ができるか」(以降, 情報提供) を評価する。模範解答プログラムは, 学習者が作成したプログラムが構文制約を満たしているかで評価する。実行例に関してはカバレッジ率を評価基準とする。

3.3 評価方法

本研究の作問学習の評価方法について説明する。各提出物の評価基準と評価方法をまとめた表を表 2 に示す。

3.3.1 問題文の評価方法

問題文は, 生成 AI に学習者が解答した問題文を与え, 評価項目を満たすかを確認する指示を与えることで評価する。

はじめに生成 AI へ与えた評価基準のプロンプトを図 4 と図 5 に示す。図 4 と図 5 は佐藤 [1] のループリックの評価項目と評価基準を与えている。評価項目の内, 単元理解に関するプロンプト(図 4) には, 出題文の構文制約も与えている。

生成 AI による評価を確認するために, 図 2 の出題文に対して, ループリックによる評価点がすべて 1 から 3 点となる問題文を 3 問作成した。例として評価がすべて 3 点となる問題文を図 3 に示す。

生成 AI に図 4 と図 5 のプロンプトと作成した問題文を与えて評価を指示したところ, 想定通りの評価点を得られないことがあった。

そこで, 最初に作成した評価のプロンプト(図 4, 図 5) に加えて図 6 に示す条件分岐の問題文のサンプルとその評価例をプロンプトとして与えたところ, 図 3 などの無限ループの問題に対して適切な評価が得られた。なお, サンプルは各項目に 2 例ずつ与えている。単元理解

あなたは学校の行事で遠足に行くので, おやつを買いに行くことにしました。おやつ代は 300 円までとします。おやつ の 値 段 を 入 力 し, 合 計 金 額 が 300 円 になる まで 無 限 に 入 力 を 受 け 付 け る よ う な プ ロ グ ラ ム を 作 成 し て く だ さ い。な お, 合 計 金 額 が 300 円 を 超 え た 場 合, 直 前 の 合 計 金 額 を 実 行 結 果 に 表 示 す る よ う に し て く だ さ い。ま た, 入 力 の 受 付 が 終 了 し た 場 合, そ の 時 の 合 計 金 額 と 実 行 例 を 表 示 し て く だ さ い。

図 3 ループリックによる評価点がすべて 3 の問題文

と創意工夫は評価点が 1 と 3 の 2 問, 情報提供は評価点が 1 と 2 の 2 問をサンプル例として与えている。単元理解に関しては, 図 6 のように構文制約も与えている。創意工夫に関しては, 実際の講義で使用された練習問題は与えず, 他 2 つの評価項目と同じサンプルとその評価例を与えたが, 想定通りの評価点が得られた。情報提供に関しては, 評価点が 3 の条件が「評価点 1 と 2 を満たしていない」なので, 評価点が 1 と 2 のサンプル例を与えている。

以下に示す C 言語プログラムの問題文についての評価を行ってください。

C 言語プログラムの問題文において, 「作成された問題は単元の理解を深めるものか」について 1-3 で評価してください。

- 1: 条件を満たしていない問題文である
- 2: 条件を満たした問題文である
- 3: 2 の内容を満たしており, かつ単元の理解の狙いに沿った追加の要素を含んだ問題文である

なお, 条件とは以下の 4 つです。

1. while(1) または for(;;) を使用して無限ループを作成すること
2. ループ内で scanf 関数を使って整数を入力させ, その数値を使用して何らかの計算や処理を行うこと
3. ある条件を満たすときに if 文でチェックし, break を使ってループを終了させること
4. ループ終了後に計算結果や処理結果を表示すること

図 4 単元理解の評価基準のプロンプト

以下に示す C 言語プログラムの問題文についての評価を行ってください。

C 言語プログラムの問題文において, 「オリジナリティ・創意工夫はあるか」について 1-3 で評価してください。

- 1: 講義で取り組んだ練習問題と同じ, もしくは少し変えただけの問題文である
- 2: 講義で取り組んだ練習問題とは異なった問題文である
- 3: 2 の内容を満たしており, 且つ独自の工夫点がある

図 5 創意工夫の評価基準のプロンプト

3.3.2 プログラムの評価方法

模範解答プログラムは, 生成 AI に構文制約を満たすかの確認を行うプロンプトを与えることで評価する。評価例として図 3 に対する模範解答プログラムが構文制約を満たすかの確認を行った結果を図 7 に示す。プログラムは出題文にある構文制約を満たしているため, 図 7 は正しい出力結果といえる。

問題文を生成 AI に与えて解答プログラムを作成させ, それが構文制約を満たすかも確認する。

まず、評価基準として、問題文のサンプルを 2 問与えます。このサンプル例は、満たしていない構文制約 (構文制約 3) があるので、評価点は 1 となります。なお、構文制約は以下の 4 つです。

サンプル例：条件分岐を行うような C 言語プログラムを作成してください。条件分岐が終了した際はその結果を表示してください。

- 構文制約：1. if 文または switch 文を使用して条件分岐を作成すること
2. 条件分岐の際は else または case を使用すること
3. 文の中で入れ子構造となるような、三分岐以上の条件分岐を行う
4. 条件分岐終了後に計算結果や処理結果を表示すること

図 6 単元理解のサンプル例のプロンプト

3.3.3 実行例の評価方法

実行例は、gcc に付属している gcov コマンドを使ってカバレッジ測定を行い、すべての命令文が実行されているかを評価する。

- 提供された C 言語プログラムが構文制約を満たしているかを確認します。

構文制約の確認:

1. 無限ループの作成:

```
while(1) {
    // ...
}
```

→ 満たしています。

2. scanf関数で整数入力:

```
scanf("%d", &price);
```

→ 満たしています。

3. if文とbreakを使用してループ終了:

```
if (price < 0) {
    break;
}
```

→ 満たしています。

4. ループ終了後に結果を表示:

```
printf("合計金額は %d 円です。\\n", total);
```

→ 満たしています。

結論:

プログラムはすべての構文制約を満たしています。

図 7 構文制約を満たしているかの出力結果

4 評価

4.1 評価方法

提案した評価方法が妥当であるかを、1 年次に C 言語を学んだ大学 4 年生 8 人を対象に作問学習を行い、確かめた。学生に対して、C 言語プログラムの作成に関する出題文として図 2 と図 8 を出題し、解答してもらった。学生は作問学習に初めて取り組むので、例として、佐藤 [1] で示された条件分岐を題材とした出題文と高評価、低評価となるような問題文を示した。なお、出題の際、佐藤 [1] のループリックは示していない。

以下の構文制約を満たす C 言語プログラムを作成するような問題文、その模範解答プログラムと動作確認に必要な実行例を解答してください。

構文制約

- 引数が文字列 (char *s や char t[] など) の関数を使用すること
- 文字列の入力を受け付けるような処理を行うこと
- 処理結果を表示すること

図 8 文字列の出題文

表 3 無限ループ 単元理解 評価結果

単元理解	著者 1	著者 2	生成 AI
問題文 1	3	2	2.2
問題文 2	2	2	2.0
問題文 3	1	1	1.0
問題文 4	1	2	2.0
問題文 5	1	2	1.6
問題文 6	1	1	1.0
問題文 7	3	3	3.0
問題文 8	1	3	2.0

表 4 無限ループ 創意工夫 評価結果

創意工夫	著者 1	著者 2	生成 AI
問題文 1	3	2	3.0
問題文 2	1	1	2.0
問題文 3	2	2	2.2
問題文 4	2	1	2.0
問題文 5	2	1	2.0
問題文 6	1	1	1.0
問題文 7	3	3	3.0
問題文 8	3	3	3.0

学生からは無限ループに関する問題文 8 問と文字列に関する問題文 7 問が提出された (1 人は文字列の問題は未提出)。

提出された問題文に対して、筆頭著者である大学院修士 1 年生 (著者 1) と共著のプログラミング教育経験のある教員 (著者 2) で評価し、生成 AI の評価結果と比較する。

4.2 評価結果

4.2.1 問題文の評価

生成 AI に 3.3.1 節で述べた条件分岐のサンプルを与えたプロンプトを与えて問題文を評価した。無限ループに関する問題文の評価結果をまとめた表を表 3、表 4 と表 5 に、文字列に関する問題文の評価結果をまとめた表を表 6、表 7 と表 8 に示す。生成 AI による評価では、各問題文に対して評価を 5 回行い、その平均を示している。

著者 1、著者 2、生成 AI による評価点がすべて一致した問題文は、無限ループに関する問題文では単元理解は 4 問、創意工夫は 3 問、情報提供は 7 問となり、文字列に関する問題文では単元理解は 4 問、創意工夫は 5 問、情報提供は 3 問となった。著者 1 と著者 2 の評価結果が一致していると生成 AI による評価も概ね一致している。

4.2.2 プログラムの評価

作成された 15 問の模範解答プログラムのうち、11 問が構文制約をすべて満たしており、無限ループに関する問題は 3 問、文字列に関する問題は 1 問構文制約を満たしていなかった。また、文字列の問題で関数を使っていないプログラムが 2 問あったが、そのうち 1 問は構文制約 1 を満たしていないのにも関わらず、満たしていると評価された。無限ループの 1 問は、問題文は

表 5 無限ループ 情報提供 評価結果

情報提供	著者 1	著者 2	生成 AI
問題文 1	3	3	3.0
問題文 2	2	1	2.0
問題文 3	1	1	1.0
問題文 4	2	2	2.0
問題文 5	3	3	3.0
問題文 6	1	1	1.0
問題文 7	2	2	2.0
問題文 8	2	2	2.0

表 6 文字列 単元理解 評価結果

単元理解	著者 1	著者 2	生成 AI
問題文 1	3	2	3.0
問題文 2	2	2	2.0
問題文 3	2	2	2.0
問題文 4	2	2	3.0
問題文 5	2	2	1.6
問題文 6	2	2	2.0
問題文 7	3	3	3.0

表 7 文字列 創意工夫 評価結果

創意工夫	著者 1	著者 2	生成 AI
問題文 1	3	2	3.0
問題文 2	1	1	1.0
問題文 3	1	1	1.0
問題文 4	1	1	1.0
問題文 5	1	1	1.0
問題文 6	1	1	2.0
問題文 7	3	3	3.0

ユーザーから自然数（1 以上の正の整数）を複数回入力させ、その合計を計算して表示するプログラムを作成しなさい。ただし、自然数以外が入力された場合はループを終了する。

であるが、模範解答プログラムでは、最初に回数を入力し、その回数分繰返し整数を入力していた。

生成 AI に問題文を与えて解答させたプログラムも評価した。無限ループに関する問題が 3 問、文字列に関する問題が 1 問構文制約を満たしていない結果になった。無限ループに関する問題は、2 問が学生が解答した模範解答プログラムと同じ構文制約（構文制約 4）を満たしていないプログラムが作成され、1 問は学生が解答した模範解答プログラムでは無限ループを使用していたものの、問題文から作成されたプログラムは無限ループを使用しないものが作成された。文字列に関する問題は、学生が解答した模範解答プログラムでは引数が文字列の関数を使用していたが、問題文から作成されたプログラムでは使用していなかった。また、作成されたプログラムの中には main 関数のみで構成されているプログラムがあったが、それらはすべて構文制約 1 を満たしていると評価された。これは構文制約 1 の文章「引数が文字列 (char *s や char t[] など) の関数を使用すること」の解釈違いによって起こった誤評価である。この制約は、文字列が引数の関数を作成して使用することを意図していたが、生成 AI は strcmp 関数などのライブラリ関数の使用でも構文制約 1 が満たされると評価した。

4.2.3 実行例の評価

15 問の実行例のうちステートメントカバレッジ 100% のものは 11 問であった。100%にならなかった問題とし

表 8 文字列 情報提供 評価結果

情報提供	著者 1	著者 2	生成 AI
問題文 1	3	1	3.0
問題文 2	1	1	1.0
問題文 3	1	1	1.0
問題文 4	3	3	3.0
問題文 5	3	3	1.8
問題文 6	3	1	3.0
問題文 7	2	1	2.0

て、scanf, fgets の入力においてエラーチェックをしているがエラーとなる実行例がない問題が 2 問あった。大文字を小文字に変換する問題で、if-else を使って分岐をしているが、else の場合の実行例がない問題が 1 問、無限ループの問題で整数の合計を計算した後、条件分岐でメッセージを変える問題で、一部の分岐の実行例がない問題が 1 問あった。また、英大文字を小文字に変換するプログラムに対して、ひらがなをカタカナにしている誤った実行例もあった。

5 考察

作問学習の感想も回答してもらったが、自分で問題を作成する機会がないので苦労したという旨の意見が多かった。問題文を見ると、無限ループに関する問題文は 8 問中 5 問が整数を繰返し入力し、合計を求めるような問題文であった。文字列の問題文は文字を数えるような問題文が 3 問、大文字を小文字に変換するような問題文が 2 問、入力された文字列を逆順に表示するような問題文が 1 問と講義で解答したような問題文が多かった。また、無限ループと比べて一文で終わるような単調な問題文が多かったことから、文字列に関する問題文作成の難しさが窺える。

評価について、生成 AI に創意工夫の評価基準を与える際、サンプル例に実際に講義で使用した問題文を与えていないが、一定の評価が得られた。実際の問題文を与えることでより適切な評価が得られるか確かめることが今後の課題である。

文字列に関する問題の評価では、生成 AI が構文制約 1 の評価をこちらの意図通りに行えなかった問題があったので、構文制約文の改善が必要である。プログラムの評価では、生成 AI に問題文のみを与えて評価を行ったが、作成された実行例も与えた場合、作成された模範解答プログラムと同じプログラムが生成されるかを確かめる。

6 おわりに

本論文では、プログラム作問学習における生成 AI を用いた評価方法を提案した。著者 2 人が共通の評価をした問題文について、約 8 割が生成 AI でも同じ評価が得られた。今後の課題としては、より多くの問題や学習者を対象にした評価や生成 AI による評価の精度向上などが挙げられる。

謝辞

本研究の一部は JSPS 科研費 23K11359、2025 年度南山大学パッへ奨励金 I-A-2 の助成を受けた。

参考文献

- [1] 佐藤晴日: プログラミング教育における作問学習による学習効果向上の定量的評価—プログラミング的思考の観点から—, 4 pages, 会津大学短期大学部産業情報学科経営情報コース 2019 年度卒業研究論文要旨集 (2019)
- [2] 大石陸斗, 高木正則: 作問学習で作成された問題の診断・評価およびフィードバックに対する対話型生成 AI の活用, 情報処理学会 情報教育シンポジウム 2024 (SSS2024), pp.250-257 (2024)