

令和 5 年 5 月 26 日現在

機関番号：12601

研究種目：基盤研究(C)（一般）

研究期間：2020～2022

課題番号：20K12106

研究課題名（和文）記号実行を用いたプログラミング試験問題の自動生成

研究課題名（英文）Automatic generation of programming questions by symbolic execution

研究代表者

萩谷 昌己（Hagiya, Masami）

東京大学・Beyond AI 研究推進機構・特任研究員

研究者番号：30156252

交付決定額（研究期間全体）：（直接経費） 3,300,000円

研究成果の概要（和文）：プログラミング初級者を対象としたプログラミング試験において、自動採点を行う手法を確立し、この手法に基づいた試験実施プロトタイプシステムを設計・実装した。従来手法では、不正解のプログラムを正解と誤判定する可能性があるが、本手法においては、初級者に典型的に出題される問題を含む一定の範囲内で、不正解のプログラムを必ず判別することが可能である。設計したシステムはブロック部品を組み合わせるものであり、出題者は部品の粒度を調整して難易度を制御することができる。

研究成果の学術的意義や社会的意義

プログラミング教育の重要性は広く認知されており、大学入学共通テストでも、プログラミングをその内容として含む教科「情報」が加えられた。本研究で確立した手法は、答案として提出されたプログラムを自動的に誤りなく採点するという特長を有しており、このような大量の答案を短期間に採点する試験に適する性格を持っている。今後、機能面などでさらなる改良を加えることで、実用性を高めることが可能である。

研究成果の概要（英文）：We have established an automatic scoring method for programming examinations for beginners and have designed and implemented a prototype system for conducting examinations based on this method. In the existing method, a wrong answer may be misjudged as a correct answer. In the proposed method, a wrong answer in a specific range of problems including those for beginners is always judged as incorrect. In the designed system, the examinee combines blocks to build a program, and the questioner can control the difficulty level by adjusting the granularity of the block parts.

研究分野：プログラミング教育

キーワード：プログラミング教育 情報教育 プログラミング言語 記号実行 定理証明 仕様記述・検証 ソフトウェアテスト

科研費による研究は、研究者の自覚と責任において実施するものです。そのため、研究の実施や研究成果の公表等については、国の要請等に基づくものではなく、その研究成果に関する見解や責任は、研究者個人に帰属します。

1. 研究開始当初の背景

AI や IoT に代表されるソフトウェア技術の急進展を背景に、高等教育のみならず、初等・中等教育においてもプログラミング教育を充実させるためのさまざまな施策が開始されつつあった。現実的な見地からは、大学入学試験にプログラミング (を含む情報分野) の問題が出題されるかどうかは、実際に学習する生徒にとっては大いに関心のあるところであり、学習のモチベーションを左右する。当時、文部科学省は、大学入試センター試験に代わる大学入学共通テストで「情報」を出題の教科に加えることを検討していた (その後、実際に加えられた)。この観点から、大学入学共通テストにも応用でき、かつ、精確な評価結果が期待できるプログラミング試験の方法を開発することは、大きな社会的意義を持つと考えられた。

上記のような大量の受験者を対象とする試験においては、自動採点が要求される。選択式の試験ではその実施方法は自明であるが、記述式の試験では、個別の工夫が必要になる。プログラムを作成させる形式の試験においては、プログラムに対する入力とそれに対する正答となる出力のペアを多数用意しておいて、受験者の作成したプログラムからの出力が、用意された出力と一致することをもって正答とする、という方法が広く用いられている。この方法は、正しい答えは必ず正しいと判断できる一方で、誤った答えも正しいと判断してしまうことがある、という問題点を持つ。

一方、ソフトウェアのテスト・検証の分野で広く用いられている技術に、記号実行がある。通常のプログラム実行が、具体的な値を用いて行われるのに対し、記号実行は変数 (記号) を含む式を値として扱い、プログラムがどのような実行経路を辿るかを分析する。SMT ソルバーによる充足可能判定を行うことで、実行経路の実現可能性などを判定することができる。

2. 研究の目的

記号実行技術を、プログラミング試験の実施に応用することをめざす。すなわち、記号実行技術が、プログラミング試験問題の採点等の鍵となる技術になり得ることを示すことが、本研究の目的である。

3. 研究の方法

プログラミング初級者を対象とする試験を、本研究の主たる対象とする。プログラミングに関する理解を問う試験には様々な形がありうるが、本研究では、そのもっとも基本的な形である、受験者にプログラムを作成させる試験を想定する。

記号実行の考え方をもとに、自動採点を行う手法を構築する。具体的には、以下を行う。

- (1) 試験に用いるプログラミング言語を設計し、処理系を実装する。
- (2) 受験者の答案の正誤を自動判定する手続きを設計・実装する。
- (3) 受験者が試験問題を解き、提出するシステムを構築する。
- (4) 上述のシステムを用いて、試験が実施できることを実験により確認する。

4. 研究成果

前節の各項目に沿って、研究成果を述べる。

(1) プログラミング言語の設計・処理系の実装

記号実行技術は、ソフトウェアの検証で用いられているが、常に検証が可能なわけではない。その鍵となるのは充足可能判定であるが、一般には充足可能判定問題は決定不能である。本研究では必ず正しい採点結果が得られることをめざすのであるから、なんらかの制約を付すことが必要になる。

そこで、Presburger 算術をその基本とする体系を構築することにした。Presburger 算術は、整数の範囲で、加法と大小比較を演算として採用する理論であり、その充足可能判定問題は決定可能であることが知られている。さらに、Z3 などの SMT ソルバーに、その決定手続きが実装されている。

受験者が提出したプログラムの正当性が、Presburger 算術の論理式の充足可能性に帰着するようなプログラミング言語を設計した。その特徴は以下の通りである。

- ・データ型は、スカラーは整数のみで、演算は加法のみである。ただし、定数倍は許す。
- ・整数の配列は使用できるが、サイズは定数に限る。
- ・条件判定が行える。
- ・定数回の繰返しが行える。break 文で脱出することもできる。
- ・while 文は、繰返し部において、変数に定数を加える演算のみが可能である。

・ assert 文, assume 文を持つ . 条件式には Presburger 論理式が指定できる .
このプログラミング言語の処理系を ,Python を用いて実装した . 構文解析には Lark を用いている .

(2) 正誤判定手続きの設計・実装

上記のプログラミング言語の記号実行系を作成した . 記号実行の一般論に従っているが , 上記の言語設計に合わせて , 以下の点に特徴を持つ .

- ・ 配列は定数長であることを利用して , 各要素ごとに (Presburger の) 変数を対応させる .
- ・ 繰返しが定数回であることを利用して , すべての実行パスを列挙する .
- ・ while 文では , 繰返し回数 t を用いて , 各変数の値が t の一次式で表されることを利用して , ループ不変式を自動生成する .

この記号実行方式に基づき , 受験者の答案を次のように採点することができる . 想定する解答 (正答) と , 受験者のプログラムに同じ入力を与え , その出力を 2 つの変数 expected と actual に代入するプログラムを生成する . 最後に , assert (expected == actual); という文を追加する . このプログラムの記号実行を行い , 挿入した assert 文が成功すれば , 受験者のプログラムは正しいと判定できる . 失敗すれば誤りであると判定できる .

以上の方式を , Python を用いて実装した . 構文解析部分は , 言語処理系と共通である . Presburger 論理式の充足可能性判定は , Z3 を用いた .

以下を確認する目的で , 実装した正誤判定手続きを用いた採点の実験を行った .

・ 提案するプログラミング言語で , プログラミング初級者に課すような典型的な試験問題の解答が記述できること

- ・ 正誤判定手続きが正しく機能し , 実用的な時間内に判定ができること .

実験の結果 , 両者ともに確認することができた . ただし , 第 1 点については , 主に定数制約を原因として , 適切な言換えが必要となる場合がある . また , 第 2 点については , 問題文中で指定する定数の大きさを適切に設定する必要がある .

(3) 解答作成・提出システム

前項の言語処理系及び判定手続きを用いれば , 受験者の解答を採点することが可能である . しかし , この言語特有の文法上の制約が , 特にプログラミング初級者にとって理解することが難しい可能性がある . このため , ブロック部品を組み合わせてプログラミングを行う環境を提供し , 受験者がこの環境を用いて試験問題に対する解答を作成・提出することができるようにした . 図 1 に , この環境を用いて作成したプログラムの例を示す .

基本的なブロック部品はシステムが用意するが , 出題者がブロック部品を作成することも想定している . このために簡易言語を設計した . 出題者はこの簡易言語を用いてブロック部品を定義し , 解答者に使用させることができる .

この解答方式は , 上述のプログラミング言語に限らず , Python などの一般的なプログラミング言語に対しても適用することができる . 以下の利点がある .

- ・ 受験者に 詳細な文法知識を要求しないようにできる .
- ・ 受験者に対して , 言語が持つ一部の機能を使用させないようにすることができる .
- ・ 受験者に入出力などに関する知識を要求しないようにできる .
- ・ ブロックの粒度を調整することにより , 試験問題としての難易度を制御することができる .

この解答作成・提出システムを , 言語処理系や正誤判定手続きと統合する形で実装した . ユーザインタフェース部分は JavaScript で記述されており , ウェブブラウザで動作する . ブロック部品は blockly を利用して作成している .

(4) 実験

上述のシステムの可用性を検証するための実験 (模擬試験) を行った . 受験者は , クラウドソ



図 1. ブロック部品によるプログラミング

ーシングサイトで募集を行った 195 名であり、主としてプログラミング初級者である。各受験者には 3 題の問題を与え、システム使用法を理解する時間も含めて 60 分以内の解答を求めた。ある問題では、受験者はランダムに 3 つのグループに分けられた。1 グループは従来方式で、Python 言語によるプログラムの作成が課された。残りの 2 グループはブロックプログラミングであるが、それぞれ、細粒度と粗粒度の部品が与えられた。

本実験を通して、以下のことが確認された。(1) 本研究の提案する自動採点方式によって、プログラミング初級者への問題の採点が可能であること。(2) Python などの既存言語で解答させる方式と同様に、ブロックプログラミング方式においてもプログラミング試験を実施できること。(3) ブロック部品の粒度を粗くすることで、問題に対するヒントを与えることができる一方で、アルゴリズムに制約を導入することから、常に問題を易しくする効果があるとは言えないこと。

(5) まとめ

本研究では、プログラムを作成させる形式の試験の自動採点方式を提案し、この方式に適した受験用プログラミングシステムを実装し、模擬試験を行う実験によってその可用性を検証した。今後、性能面などでさらなる改良を加えることにより、大量の答案を短期間に採点する試験への適用が期待できる。

5. 主な発表論文等

〔雑誌論文〕 計5件（うち査読付論文 2件/うち国際共著 0件/うちオープンアクセス 0件）

1. 著者名 Yoshinori Tanabe, Masami Hagiya	4. 巻 LAIS 30
2. 論文標題 Automatic Scoring in Programming Examinations for Beginners	5. 発行年 2022年
3. 雑誌名 Knowledge-Based Software Engineering: 2022 - Proceedings of the 14th International Joint Conference on Knowledge-Based Software Engineering	6. 最初と最後の頁 127-139
掲載論文のDOI（デジタルオブジェクト識別子） 10.1007/978-3-031-17583-1_10	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -

1. 著者名 田辺良則, 萩谷昌己	4. 巻 2023-CE-168
2. 論文標題 ビジュアルプログラミングによるプログラミング試験について	5. 発行年 2023年
3. 雑誌名 情報処理学会研究報告コンピュータと教育	6. 最初と最後の頁 1-7
掲載論文のDOI（デジタルオブジェクト識別子） なし	査読の有無 無
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -

1. 著者名 萩谷昌己	4. 巻 14
2. 論文標題 情報教育課程の設計指針と大学入学試験	5. 発行年 2021年
3. 雑誌名 日本情報科教育学会誌	6. 最初と最後の頁 5-12
掲載論文のDOI（デジタルオブジェクト識別子） なし	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -

1. 著者名 田辺良則, 萩谷昌己	4. 巻 121
2. 論文標題 初級者を対象としたプログラミング試験のためのシステムについて	5. 発行年 2022年
3. 雑誌名 信学技報	6. 最初と最後の頁 7-11
掲載論文のDOI（デジタルオブジェクト識別子） なし	査読の有無 無
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -

1. 著者名 田辺良則, 萩谷昌己	4. 巻 120
2. 論文標題 試験問題の自動採点が行えるプログラミング言語の検討	5. 発行年 2021年
3. 雑誌名 信学技報	6. 最初と最後の頁 48-53
掲載論文のDOI (デジタルオブジェクト識別子) なし	査読の有無 無
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -

〔学会発表〕 計4件 (うち招待講演 0件 / うち国際学会 1件)

1. 発表者名 Yoshinori Tanabe, Masami Hagiya
2. 発表標題 Automatic Scoring in Programming Examinations for Beginners
3. 学会等名 14th International Joint Conference on Knowledge-Based Software Engineering (JCKBSE 2022) (国際学会)
4. 発表年 2022年

1. 発表者名 田辺良則, 萩谷昌己
2. 発表標題 ブロック部品を用いたプログラミング試験について
3. 学会等名 情報処理学会 コンピュータと教育研究会 168回研究発表会
4. 発表年 2023年

1. 発表者名 田辺良則, 萩谷昌己
2. 発表標題 初級者を対象としたプログラミング試験のためのシステムについて
3. 学会等名 電子情報通信学会知能ソフトウェア工学研究会
4. 発表年 2022年

1. 発表者名 田辺良則, 萩谷昌己
2. 発表標題 試験問題の自動採点が行えるプログラミング言語の検討
3. 学会等名 電子情報通信学会知能ソフトウェア工学研究会
4. 発表年 2021年

〔図書〕 計0件

〔産業財産権〕

〔その他〕

-

6. 研究組織

	氏名 (ローマ字氏名) (研究者番号)	所属研究機関・部局・職 (機関番号)	備考
研究分担者	田辺 良則 (Tanabe Yoshinori) (60443199)	鶴見大学・文学部・教授 (32710)	
研究分担者	斎藤 俊則 (Saito Toshinori) (80434447)	星槎大学・教育実践研究科・准教授 (30124)	

7. 科研費を使用して開催した国際研究集会

〔国際研究集会〕 計1件

国際研究集会 IFIP WCCE 2022: World Conference on Computers in Education	開催年 2022年～2022年
--	--------------------

8. 本研究に関連して実施した国際共同研究の実施状況

共同研究相手国	相手方研究機関
---------	---------