

Scratch ブロックの多様な組み合わせを考慮した プログラム自動採点方法の開発

畠中 明哉^{*1}, 広瀬 啓雄^{*1}, 尾崎 剛^{*1}

^{*1} 公立諏訪東京理科大学大学院

Development of an Automated Grading Method for Scratch Programs Considering Diverse Combinations of Blocks

Akiya HATANAKA^{*1}, Hiroo HIROSE^{*1}, Takeshi OZAKI^{*1}

^{*1} Suwa University of Science

This study developed an automated grading system for Scratch programs. The system extracts the required information from the solution programs, compares them with the model answers to determine correctness, and displays the results. By validating the constrained problems with multiple expected answers, it was confirmed that accurate grading is possible.

キーワード: プログラミング教育, ビジュアルプログラミング, Scratch, 初等教育

1. はじめに

日本では, IT 人材不足の解消を目的に, 2020 年度から小学校でのプログラミング教育が導入された. 文部科学省が公開している「小学校プログラミング教育の手引(第三版)」¹では, 「プログラミング的思考」の育成や各教科等での学びをより確実なものにすることなどがねらいとして挙げられている. 小学校のプログラミング教育は, 多様な教科等に取り入れることが可能であり, そのためには児童がプログラミングに必要な知識・スキルを身につけるための学習活動を行う必要がある. しかし, その学習活動を各教科等に組み込んでしまうとプログラミング教育の本来の目的が達成されない可能性がある. そのため, 教員が授業時間をやりくりしてプログラミングを学ぶ時間を確保しているが, 十分な時間を確保することができない可能性がある. そこで, 児童が短時間で効率的な学習を可能にし, どの程度プログラミング的思考やプログラミングに関する知識・スキルを習得したか確認するための方法を明らかにする必要がある.

Scratch などのビジュアル型プログラミング言語を用いたプログラムを対象に, 学習者が作成したプログラムに対して自動で分析や評価を行う研究は既に行われている. 太田ら²は, 小学校におけるプログラミング教育での教師や児童の利用を想定しており, 児童が自由に作成したプログラムに対して, 分析や評価を行うための基準を新たに提案し, その基準に基づいてプログラムの評価を行う学習支援システムを開発した. このシステムはプログラムの評価を行うものであり, 特定の課題に対して児童が解いた問題の自動採点を行うシステムではない.

著者らは, プログラムの効率的な学習用教材として児童が数分程度取り組める Scratch ドリルの開発を目指している. これまで, プログラミングを学ぶ課題に対して, 作成されたプログラムを正しく採点する正誤判定方法の提案と検証を行ってきた³. しかし, 正しく判定を行うことができなかったブロックの組み合わせが存在したため採点方法の再検討が必要であった.

これらをうけて本研究は, 学習者が問題に対して作成したプログラムを正しく判定できる正誤判定システ

ムの構築を目的に行った。

2. プログラム自動採点システム

2.1 Scratch

現在、小学校では Scratch などのビジュアル型プログラミング言語を使った学習指導が展開されている。

Scratch は、米国のマサチューセッツ工科大学メディアラボと Scratch 財団が共同開発したビジュアル型プログラミング言語である。ブロックタイプのコマンドを組み合わせることでプログラムを作成することができる。Scratch 開発環境の画面は大きく 3 つに分けられている。画面左はブロックパレット、画面中央にスクリプトエリア、画面右がステージ及びスプライト情報が表示されている。ブロックパレットには、動き・見た目・音・イベント・制御・調べる・変数・ブロック定数の 8 種類に分けられたブロックがある。これらのブロックをスクリプトエリアにドラッグ&ドロップし、組み合わせることでプログラムを作成する。プログラムの実行結果がステージ及びスプライトに表示される。

さらに、Scratch にはプログラムを拡張できるものとして拡張機能がある。2019 年に公開された Scratch3.0 では、micro:bit や翻訳、音声合成などの外部機器や API を扱う拡張機能が追加された。また、拡張機能は Scratch 公式が用意しているものの他に、拡張機能を自作することができ、Stretch3⁴ のような第三者が作成した拡張機能を搭載した Scratch システムが Github 上で公開されている。本研究では、Scratch3.0 をつかったシステムを構築し、以下、Scratch3.0 を Scratch と表記する。

2.2 自動採点の流れ

自動採点は Scratch の拡張機能を用いて実現する。図 3 はプログラムの自動採点の流れである。学習者が拡張機能を選択すると、図 4 のように Scratch のブロックパレットに問題ごとに作成された正誤判定ブロックが追加される。問題に対する解答を作成した後、対応する正誤判定ブロックを先頭に配置することで正誤判定が行われる。正解不正解の判定結果がステージ上

に表示される。

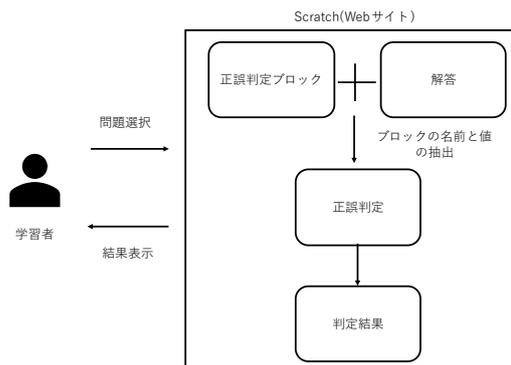


図 1 自動採点の流れ

2.3 採点機能の構築

Scratch のソースコードは、GitHub 上で公開されている。公開されているソースコードのうち、scratch-gui と scratch-vm をダウンロードし、拡張機能の構築に関係のあるファイルを編集することで、拡張機能を自作することができる。本研究では、公式の Japanese Scratch-Wiki⁵ の拡張機能の作り方を参考に、新たな拡張機能として、正誤判定を行うオリジナルブロックを作成していく。また、Scratch の開発言語は JavaScript であるため、オリジナルブロックの動作を記述する際も JavaScript を使用する。

表 1 ブロックの種類と形状

種類	形状
真偽ブロック	
スタックブロック	
ハットブロック	
C型ブロック	
値ブロック	
キャップブロック	

Scratch で利用可能なブロックを形状別でみた場合、表 1 に示す 6 種類がある。これらのブロックのうち、ハットブロックは、その下にブロックが追加される度

に処理が行われる。この特徴は、学習者が作成したプログラムの正誤判定を行うのに適しているため、本研究では正誤判定ブロックをハットブロックで作成した。また、値ブロックの中には、選択するとそのブロックに代入されている値をステージ上に表示することができるブロックがある。このことから、判定結果の表示は値ブロックで作成した(図2)。



図2 正誤判定ブロック・判定結果ブロック

2.4 正誤判定

正誤判定は、解答プログラムに使われたブロック名、値、順序が模範解答と一致するかどうかで行う。

2.4.1 Scratch のブロック

学習者が作成したプログラムを採点する上で、Scratch のブロックから必要な情報を読み取る必要がある。Scratch にはブロックパレット、スクリプトエリアにある全ブロックの情報をもつ BlockUtility オブジェクトというプロパティがあり、ステージ上のスプライトの座標、向き、大きさやスプライトの動きのメソッドなどの情報をもっている。スクリプトエリアに作成したプログラムの情報も BlockUtility に存在する。BlockUtility からスクリプトエリアに作成したプログラムを解析すると、どのブロックが使用された読み取ることができる。また、ブロックにはブロックパレットに置かれているブロックの他にも、値を入力・変更することができるブロックなどは、隠しブロックと呼ばれる値を格納しているブロックが存在する。

Scratch のブロックには、id・opcode・inputs・

fields・next・topLevel・parent・shadow・x. y というプロパティが格納されている。この中で正誤判定に必要なプロパティは id・opcode・inputs・fields・next プロパティである。id プロパティはスクリプトエリアに置かれているブロックに振り分けられている独自の id である。opcode プロパティはブロック名を表しており、例えば、「○歩動かす」ブロックの opcode は、「motion_movesteps」である。inputs プロパティは隠しブロックが存在するブロックや、「ずっと～する」や「もし～なら」などの繰り返し処理・条件分岐処理のブロックがもっており、隠しブロックの値や繰り返し・条件分岐処理のブロックの条件やその中に置かれているブロックを知りたい場合は inputs を調べると分かる。fields プロパティは隠しブロックの値が格納されており、入力された値を調べる際に有効である。next プロパティは、次に置かれているブロックの id をもっており、ブロックの並び順は next をたどることで分かる。

2.4.2 解答プログラムの読み取り

解答プログラムからブロック名、値、順序を自動的に読み取る機能を構築する。学習者が作成したプログラムからブロック名と入力された値を抜き出し、ブロック名を key、入力された値を value としたプロパティを、オブジェクトに格納する。ブロックの名前は opcode を、入力された値は隠しブロックの fields を使う。



図3 「○歩動かす」ブロック

図3の「○歩動かす」ブロックを例に説明する。「○歩動かす」ブロックの opcode は「motion_movesteps」で、入力されている値は 10 である。これを {"motion_movesteps":10} としてオブジェクトに格納する。



図4 多様なブロックの組み合わせ

しかし、Scratch では多様なブロックの組み合わせで同じ動きをすることが可能である。図4は、図3のブロックと同じ動きをするブロックの組み合わせの例で、(1)ブロックを複数置いた組み合わせ、(2)「〇回繰り返す」ブロックを使用した組み合わせ、(3)演算ブロックを使用した組み合わせである。これらも図3の場合と同様に、{"motion_movesteps":10}となるようにした。



図5 プログラム例1

図5は、「動く・待つ・動く」のような複数の動作をするプログラムの例である。このようなプログラムでは、ブロックの順序を取得する必要がある。ブロックの順序は、ブロックの next プロパティを辿ることで取得することができる。図5のプログラムは、間に違う種類のブロックを挟んで、同じ種類のブロックが配置されている。これらのブロックの順序を明確にするために、opcode に番号をつけて格納する(図5右)。

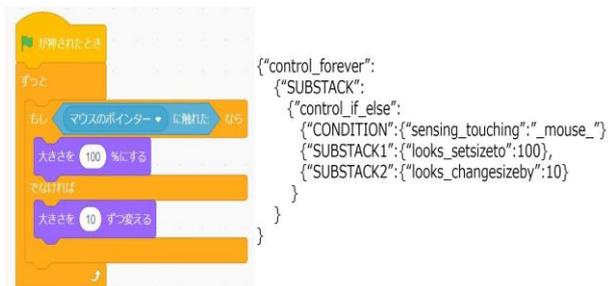


図6 プログラム例2

図6は、繰り返し処理と条件分岐処理のブロックが組み込まれたプログラムである。繰り返しや条件分岐などの制御ブロックの中などに置かれているブロックは、inputs プロパティを使って読み取ることができる。「control_forever」は、「ずっと～する」ブロックを指している。このブロックのように中に置けるブロック群が1つの場合は、inputs.SUBSTACK プロパティを調べることでプログラム群の読み取りが可能となる。

「control_if_else」は「もし～なら、でなければ～」ブロックを指す。このブロックは条件ブロック、条件が真の場合のブロック群、および条件が偽の場合のブロック群を置くことができる。これらは、「ずっと～する」ブロックと同様に inputs プロパティを使って調べる。inputs.CONDITION プロパティから条件ブロック、inputs.SUBSTACK1 プロパティから真の場合のブロック群、inputs.SUBSTACK2 プロパティから偽の場合のブロック群を読み取ることが可能である。

以上のような処理を行うことで、解答プログラムから必要な情報だけを抜き出し、オブジェクトに格納することができる。

2.4.3 正誤判定

正誤判定のために模範解答を用意する。問題に対する模範解答プログラムを上記の方法でオブジェクトに格納したものを模範解答とする。

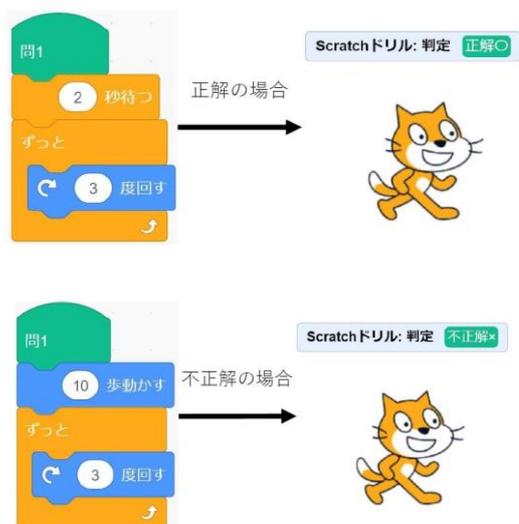


図7 判定結果の表示

図7のように正誤判定ブロックに解答プログラムを

組み合わせることで、解答が模範解答と一致するかどうかを判定する。解答が模範解答と一致していれば「正解○」、そうでなければ「不正解×」と表示する。

3. 検証

検証用の問題とそれに対する正誤判定ブロックを 7 つ作成し(表 2), 正しく判定を行うことができるかどうかを検証した。

表 2 検証用の問題

	問題文
問 1	1 秒待った後に、時計回りに 3 度回し続けなさい。
問 2	もしスプライトがマウスポインターに触れたなら 3 歩動かさなさい。そうでなければ 2 歩動かさなさい。そして、もし端に着いたら跳ね返るようにしなさい。これらをずっと繰り返しなさい。
問 3	スプライトの大きさを 100%にしなさい。そして、スプライトの大きさを 10 ずつ変え、もしマウスポインターに触れたなら大きさを 100%にする動きをずっと繰り返しなさい。
問 4	スプライトの大きさを 100%にしなさい。そして、1 秒待ちスプライトの大きさを 5 ずつ変え、1 秒待ち・スプライトの大きさを 50 ずつ変える動きをずっと繰り返しなさい。
問 5	1 秒待ち、スプライトを 0 度に向けなさい。次に 1 秒待ち、スプライトを 90 度に向けなさい。
問 6	スプライトを時計回りに 15 度回し 1 秒待ち、時計回りに 90 度回し 1 秒待つ動きを 3 回繰り返しなさい。
問 7	スプライトがマウスポインターに触れている時はずっと時計回りに 5 度回しなさい。

検証用の問題は、1 つのスプライトかつ 1 つのスク립トでプログラムを作成できる問題に限定し、「プログラミング能力検定 過去問題集 ビジュアル言語版 レベル 1」⁶の 6 月分の問題を参考に、プログラムの基

本制御構造である、順次・分岐・反復を含む問題となるように作成した。

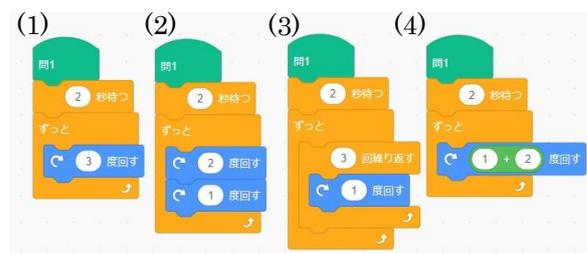


図 8 正解の想定解答

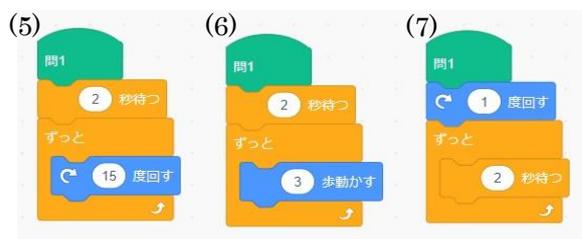


図 9 不正解の想定解答

検証用の問題に対して、正解の解答には、図 8 のような (1)「模範解答」、(2)「ブロックを複数置いた解答」、(3)『『○回繰り返す』ブロックを置いた解答』、(4)「演算ブロックを用いた解答」、不正解の解答には、図 9 のような、(5)「数値が異なるブロックを置いた解答」、(6)「ブロックの種類が異なる解答」、(7)「ブロックの順番が異なる解答」の想定解答を用意した。(2)から(7)の想定解答においては、ブロックの数値やブロックの組み合わせのパターンを複数用意した。これらの想定解答を用いてシステムの検証を行った。その結果、すべての問題に対し、用意したすべての想定解答で正しく判定を行うことができた。

4. 考察

本研究では、1 つのスプライトかつ 1 つのスク립トで作成することができる 7 つの問題で検証を行った。その問題に対し、正解・不正解の想定解答を複数用意し検証を行った結果すべて正しく判定を行うことができた。このことから、構築した正誤判定システムは自動採点が可能であるといえ、この条件に合致するプログラムを学ぶ際に利用可能であると考えられる。

Scratch のプログラムには、検証問題のようなプロ

グラム以外にも、変数を利用するプログラムや、1つのスプライトに対して2つ以上のスクリプトで作成するプログラム、複数のスプライトに対して1つまたは複数のスクリプトで作成するプログラムもある。今回構築したシステムはこれらのプログラムに対応しておらず、このようなプログラムでの検証は行っていないが、これらに対応した正誤判定システムの実現は十分に可能であると考え、開発を進めている。

5. まとめ

本研究は、Scratch で作成されたプログラムの自動採点システムの開発を行った。解答プログラムから、正誤判定に必要なブロック名、値、順序を抜き出す機能を構築した。その抜き出した情報が、模範解答と一致しているかどうかを判定し、結果を表示するシステムを開発した。

本システムで正しく採点を行うことが可能か検証するために、1つのスプライトかつ1つのスクリプトでプログラムを作成することができるプログラムに限定した問題を7つ作成した。その問題に対し、正解・不正解の想定解答を複数用意し検証を行った結果すべて正しく判定を行うことができた。

しかし、今回の研究では、限定された条件下の問題であり、変数を利用するものや2つ以上のスクリプトを使うものなどで正しく判定を行うことができるか検証できていない。

児童が短時間で効率的に学びことができる教材として Scratch ドリルシステムの開発し、活用してもらうためには、小学生が学ぶべきプログラミング要素を整理し、各要素を段階的に学習できる問題を作成する必要がある。問題の作成にあたっては、実際に小学生に取り組んでもらい、難易度・学習効果・問題の受容性について検証を行う。最終的には Scratch ドリルシステムを小学生に使用して学習を行ってもらい、プログラミング的思考やプログラミングに関する知識・スキル習得度合いを確認するための方法として有効であるか検証する予定である。

謝辞

本研究は、JSPS 科研費 22K02893, 23K02715 の助成を受けたものです。

参考文献

- (1) 文部科学省:「小学校プログラミング教育の手引(第三版)」,(2020)
- (2) 太田 剛, 加藤 浩, 森本 容介:“コンピューショナル・シンキング概念に基づくプログラム自動評価機能を持つ Scratch 用学習支援システムの開発”, 教育システム情報学会誌, Vol. 35, No. 2, pp204-214, (2018)
- (3) 畠中 明哉, 尾崎 剛, 広瀬 啓雄:“Scratch ドリルシステム開発に向けた Scratch プログラム自動採点方法の検討”, 教育システム情報学会 2022 年度学生研究発表会 A06, (2023)
- (4) Stretch3,
<https://github.com/stretch3/stretch3.github.io>
(2023 年 6 月 13 日確認)
- (5) Scratch3.0 の拡張機能を作ってみよう-Japanese Scratch-Wiki,
https://ja.scratch-wiki.info/wiki/Scratch_3.0の拡張機能を作ってみよう(2023 年 6 月 13 日確認)
- (6) プログラミング能力検定協会 著, 『2022 年度版 プロ検® 過去問題集 ビジュアル言語版 レベル 1』, 株式会社スプリックス, 東京都, (2022)