

# 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

## Examination of Scratch Program Automatic Grading Method Toward The Development of A Scratch Drill System

畠中 明哉, 尾崎 剛, 広瀬 啓雄

Akiya HATANAKA, Takeshi OZAKI, Hiroo HIROSE

公立諏訪東京理科大学 工学部 情報応用工学科

\*1Department of Applied Information Engineering, Suwa University of Science

Email: T119109@ed.sus.ac.jp

**あらまし**：本研究は、小学生が数分程度で取り組める Scratch ドリルシステムの開発に向けて、問題に対して多様なブロックの組み合わせでも正しく採点ができることを目的とする。Scratch のプロパティを取得して、ブロックの順番と入力された値から採点するシステムを構築した。ドリルで使用する予定の複数のカテゴリを含む問題に対し、正解、不正解の想定解答でシステムの検証を行ったところ、正しく正誤判定をすることができた。

**キーワード**：プログラミング教育、プログラミング的思考、Scratch、個別最適化された学び

### 1. はじめに

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。文部科学省は「小学校プログラミング教育の手引(第三版)」<sup>(1)</sup>において、「プログラミング的思考」の育成をねらいの1つとし、小学校でのプログラミング教育で使用するプログラミング言語として、ビジュアル型プログラミング言語を活用した学習指導が展開されることを想定している。また、教員は複数あるプログラミング言語や教材の中から学習内容や学習活動、児童の発達段階等に応じて、適切なものを選択し活用することが求められている。

また、個別最適化された学びの実現が求められている。個別最適化された学びとは令和元年に文部科学省が打ち出した GIGA スクール構想の1つで、目指すべき次世代の学校・教育現場として掲げた教育のスタイルを指す。生徒一人一人の理解状況や能力・適正に合わせた個別最適化された学びを行うことで、多様な子供たちが誰一人取り残されることがないようにすることが目的である。この実現のためには、児童の学習の進捗度合いなどを把握する必要があるが、そのような教材は開発されていない。

Scratch<sup>(2)</sup>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を分析や評価を行う研究が行われている。太田らは、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た、児童が自由に作成したプログラムに対して分析と評価する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<sup>(3)</sup>。この研究ではプログラムの評価を行うシステムであり、児童が解いた問題の自動採点を行うシステムではない。

そこで本研究では、児童が各自で Scratch を用いたプログラミング学習を進められるように、作成した

プログラムが自動的に採点される Scratch ドリルの開発を目指し、問題の条件を満たしている、かつ多様なブロックの組み合わせでも、正しく判定することができる採点方法の構築を目的とする。

### 2. システム概要

#### 2.1 システム概要

本研究では、Scratch3.0 の拡張機能としてオリジナルブロックである正誤判定ブロックを新たに実装することで、学習者が作成したプログラムを自動採点するシステムを構築する。図1のように、学習者は問題を選択し、既存のブロックで解答プログラムを作成する。解答プログラムの先頭に正誤判定ブロックを組み込む組み合わせることで正誤判定を行う。判定結果はスプライトに吹き出しでテキストを表示させる。

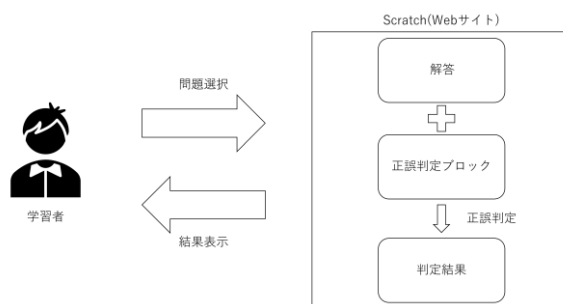


図1 正誤判定の流れ

#### 2.2 正誤判定方法

正誤判定は、解答プログラムに使用されたブロックのプロパティを取得し、分析することで行う。分析対象とするのはブロックの順番と値で、これらの2つが問題の条件を満たしていれば「正解○」、そう

でなければ「不正解×」と判定する。

ブロックの順番の判定にはブロックが持っている next プロパティを使用する。next プロパティでは、次に置かれているブロックの id を知ることができ、それを辿ることでブロックの順番を知ることができる。しかし、「ずっと～する」ブロックなど、制御ブロック内に置かれているブロックの順番を取得するには、next プロパティを辿るだけではできない。そこで、制御ブロック内に置かれている順番を取得できなかったブロックの判定には、ブロックの動きの処理に util という引数が使われているので、それを利用する。それらのブロックを実行しているときは true、実行されていないときは false とすることで、問題の条件を満たしているかの判定ができる。また、util という引数には変数を追加することができる。

「○歩動かす」ブロックのような数値の取得が必要なブロックには、util に変数を追加し、数値の取得を行う。

### 3. システムの検証

#### 3.1 検証方法

検証用問題とそれに対する正誤判定ブロックを 7 つ作成し、それぞれの問題に対する正解の解答と不正解の解答を複数用意し、正しく判定できるかを検証する。検証用問題は、「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」<sup>(4)</sup>の 6 月分の問題を参考にして、1つのスプライトかつ1つのスクリプトで解答が作成できる問題に限定する。

正解の解答として、条件を満たす最低限のブロックのみで作成された「模範解答」(図 2 (a))の他に、条件を満たすが使用しなくてもよいブロックを使用した「別解」(図 2 (b))、不正解の解答として、「ブロックに入力された値が間違っている解答」(図 2 (c))、「ブロックの種類が異なる解答」、「ブロックの順番が異なる解答」を想定解答としてシステムの検証を行う。



図 2 想定解答の例

#### 3.2 検証結果

検証の結果、想定解答は概ね正しく判定することができた。しかし、「別解」のうち「○回繰り返す」ブロックを使用した解答と、不正解の「ブロックの

順番が異なる解答」は、正しく判定を行うことができなかった。

### 4. 考察

想定解答のうち、正しく判定を行うことができなかったものの原因としては、ブロックの順番の取得方法が挙げられる。本研究では、ブロックの next プロパティを辿ることでブロックの置かれている順番を取得した。しかし、制御ブロック内に置かれているブロックは、next プロパティを辿るだけでは順番を取得することはできなかった。制御ブロック内に置かれているブロックの id は、inputs プロパティにある SUBSTACK プロパティの block プロパティが持っていることが分かった。その id の next プロパティを辿ることで、制御ブロック内に置かれているブロックの順番を取得することができるのではないかと考えられる。

また、「○歩動かす」ブロックなど、数値を入力できるブロックは、「math\_number」などの隠しブロックを持っており、それらには、fields プロパティの value プロパティに、入力された値があることが分かった。今回の数値の正誤判定方法では、util という引数に変数を追加することで、値の取得を行ったが、「math\_number」から値を取得することが出来れば util に変数を追加することなく判定ができる可能性がある。

### 5. まとめ

本研究は、Scratch ドリルの開発を目指し、Scratch で作成されたプログラムの正誤判定方法の提案と検証を行った。その結果、多様なブロックの組み合わせでも概ね正しく判定することができたが、正しく判定することができなかった解答も存在した。正しく判定できなかった原因として、制御ブロック内のブロックの順番の取得に問題があることがわかった。

今回は 1 つのスプライトかつ 1 つのスクリプトに限定したが、今後は複数のスプライトやスクリプトでも判定を可能にする必要がある。また、個別最適化された学びの実現に向けて、学習者の学習状況を取得し、分析・可視化する方法について研究を進めていきたいと考えている。

#### 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) Scratch<<http://scratch.mit.edu/>>, 2022 年 2 月 3 日確認
- (3) 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, pp.204-214, (2018)
- (4) 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