

3E-01

## プログラミング学習支援のための問題自動生成機能に関する検討

飯島 安恵\*

今野 将†

千葉工業大学大学院 工学研究科 工学専攻 博士後期課程

千葉工業大学 工学部 電気電子情報工学科

## 1 はじめに

プログラミング教育は中学において必修化されるなど、重要な教育科目の一つである。プログラミング教育は、実際にプログラムのコードを記述して学ぶ演習一体型の講義が行われる事が多い。これは、特定のプログラミング言語を学ぶ上で重要であり効果が高い学習方法であるためと考える。しかし、プログラミング教育における問題点として「学習者へ多くのコードを書かせてそれを採点するため教員への負荷が高い」「教員側が想定した難易度が学習者にとって最適とは限らない」などがあげられる。

それらの問題点を解決するために、本研究では教員の経験的知識に基づく問題の自動生成ルールを用いた、Web ベースの問題自動生成・自動採点機能付きプログラミング言語学習支援システムの提案・開発を行う。本稿では、提案システム内の「完成したソースコードから解答者の習熟度に応じた問題を自動生成する機能」を実現するにあたり、完成コードのどの部分を問題として出題するかを検討するため、問題の難易度についての調査と考察を行う。

## 2 関連研究と既存のプログラミング言語学習支援システム

プログラミング教育は中学において必修化されるなど、非常に重要な教育の一つであり、多くのプログラミング教育支援ツールが国内外で提案・開発されている。関連研究として小暮らの研究 [1] を、既存のプログラミング教育支援ツールとして、LOGO [2], ドリトル [3], Scratch [4], AIZU ONLINE JUDGE [5] を以下に示す。

小暮らの研究では、学生が何のプログラミングの課題を行っているかをリアルタイムに監視し、教員が適切な指示を与え易くするようなモニタリングシステムの開発を行っている。

LOGO とは、児童の思考能力向上の訓練を目的として開発されたプログラミング言語であり、関数型プログラミング言語である LISP を原型としている。“ロゴ坊”というソフトウェアを用いて、日本語で命令を書き図形などを出力することができる (図 1: 右)。

ドリトルとは、初心者でもプログラミングを学びやす

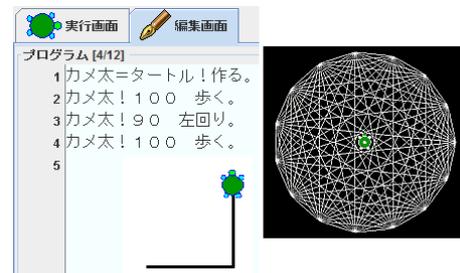


図 1: 左:ドリトル, 右:LOGO 坊 実行画面例



図 2: Scratch 編集画面例

いように設計された教育用プログラミング言語である。Web ブラウザでプログラムを実行することのできるオンライン版 (図 1: 左) と、パソコンにソフトウェアをインストールするローカル版の 2 種類があり、日本語を用いて命令を書く。

Scratch とは、言語非依存型 (ビジュアルプログラミング言語) のプログラミング言語である。Web ブラウザ上でプログラムを実行することのできるオンラインエディター (図 2) と、パソコンにソフトウェアをインストールするオフラインエディターの 2 種類がある。制御文や変数等が記された異なる形の視覚的 GUI (ブロック) を組み合わせてコードを作成する事ができる。

AIZU ONLINE JUDGE とは、会津大学が主催しているオンラインジャッジである。オンラインジャッジとは、与えられた問題を解き、ソースコードを提出しそれが正しいかどうかを判断する Web サービスである。AIZU ONLINE JUDGE では、Web ブラウザ上に用意されたエディターを用いてプログラムを書き、提出をする。

しかし、既存のプログラミング教育支援ツールの多くは導入教育向けである。そのため、導入後のプログラミング教育は依然として専門的な知識が必要である。AIZU ONLINE JUDGE では、ソースコードの採点に関しては自動で行っているが、問題の作成に関しては自動で行

Research on problem automatic generation function for programming learning support

\*Yasue Iijima, Graduate School of Engineering Doctoral Program in Engineering, Chiba Institute of Technology.

†Susumu Konno, Department of Electrical, Electronics and Computer Engineering, Chiba Institute of Technology.

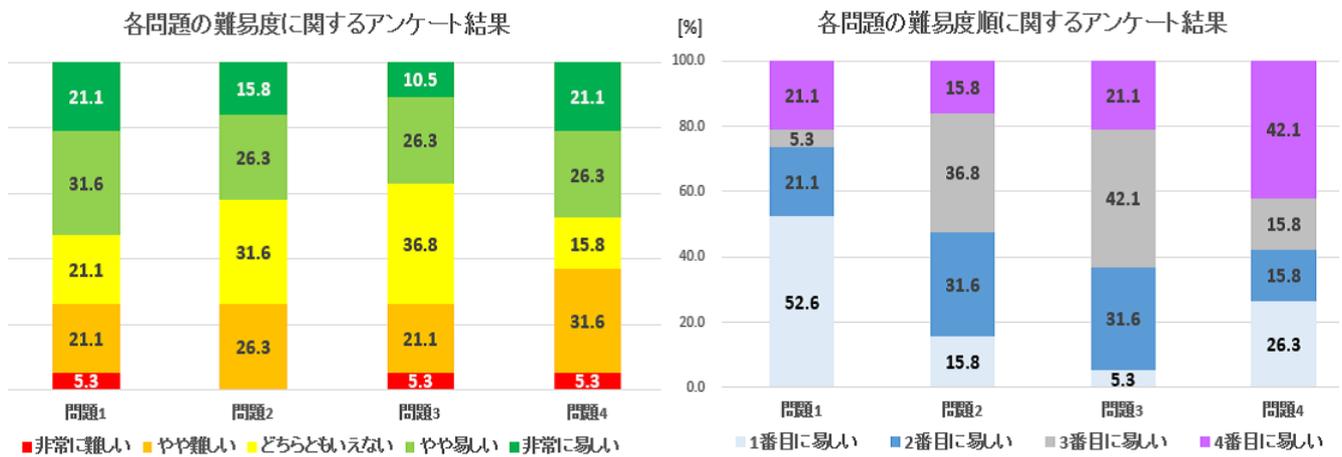


図 3: 出題箇所による難易度の違いに関するアンケート調査 結果

われていないため、問題を作成するにあたり「教員側が想定した難易度が学習者にとって最適とは限らない」という問題点が依然として発生する。

そこで、本稿では提案システム内の「完成したソースコードから解答者の習熟度に応じた問題を自動生成する機能」を実現するにあたり、完成コードのどの部分を問題として出題するかを検討するため、プログラミングの問題の難易度についての調査を行った。

### 3 問題の難易度の調査と考察

提案システムの「完成したソースコードから解答者の習熟度に応じた問題を自動生成する機能」を実現するにあたり、出題する問題の難易度について2種類のアンケート調査を行った。1つ目のアンケート調査は被験者19名に対して、出題箇所による難易度の違いに関して、2つ目のアンケート調査は被験者25名に対して、出題箇所の個数による難易度の違いに関して行った。なお、これらのアンケートの元となる問題はC言語の二次元配列の走査に関する問題を用いた。

1つ目のアンケート調査では、問題を4問用いた。結果を図3に示す。それぞれの問題の難易度を5段階で評価を行う「各問題の難易度に関するアンケート調査」では、ほぼ全ての問題の難易度について「非常に難しい」～「非常に易しい」と回答があった。4つの問題を難易度が低い順に並び替える「各問題の難易度順に関するアンケート調査」では、全ての問題に関してそれぞれの順位に振り分けられていた。どちらのアンケート調査に関しても、問題作成者側の予想とは異なる難易度、難易度順の回答が得られた。

2つ目のアンケート調査の結果では、問題を10問用いた。傾向として、出題箇所の個数が多く範囲が広い問題ほど難易度が高く、出題箇所の個数が少なく範囲が狭い問題ほど難易度が低いという傾向の回答が得られたが、

どの問題に関しても、問題作成側の難易度の想定とは異なる難易度の回答結果が得られた。これは、各学習者のC言語の習熟度が関係していると考えられる。

### 4 おわりに

本研究では、プログラミング教育における問題点である「学習者へ多くのコードを書かせてそれを採点するため教員への負荷が高い」「教員側が想定した難易度が学習者にとって最適とは限らない」を解決するために、Webベースの問題自動生成・自動採点機能付きプログラミング言語学習支援システムの提案・開発を行った。提案システム内の「完成したソースコードから解答者の習熟度に応じた問題を自動生成する機能」を実現するにあたり、完成コードのどの部分を問題として出題するかを検討するため、問題の難易度についての調査を2種類行った。結果、どのアンケート調査に関しても問題作成者側の予想とは異なる難易度・難易度順の回答が得られた。

今後の課題として、C言語の問題のバリエーションを増やすこと、学習者のプログラミング言語の習熟度に関する調査と、問題の自動生成のルールの検討が挙げられる。

### 参考文献

- [1] Satoru Kogure et al. Monitoring system for the effective instruction based on the semi-automatic evaluation of programs during programming classroom lectures. Research and Practice in Technology Enhanced Learning (2015) 10:18, pp.1-12.
- [2] LOGO, Logo Foundation (online), (<http://el.media.mit.edu/logo-foundation/index.html>) (accessed 2016-01-05).
- [3] プログラミング言語「ドリトル」, 兼宗進 (online), (<http://dolittle.eplang.jp/>) (accessed 2016-01-05).
- [4] Scratch, MIT メディアラボ ライフロングキンダーガーテングループ (online), (<https://scratch.mit.edu/>) (accessed 2016-01-03).
- [5] AIZU ONLINE JUDGE, AIZU Competitive Programming Club (ICPCPC), Database Systems Lab. (online), (<http://judge.u-aizu.ac.jp/onlinejudge/>) (accessed 2016-01-05).