5ZB-02

GitHubのプルリクエストを用いたプログラミング課題自動生成システムの実現可能性に関する検討

柴藤大介 †　矢谷浩司 †

東京大学大学院工学系研究科電気系工学専攻 †

1　はじめに

プログラミング授業の様々な教育現場への導入事例に象徴されるように，基本的な情報技術に関する適切な知識を習得させることは，教育現場における重要な課題となっている．これまでに情報教育を効率的に提供するための数多くの研究がなされてきた．例えば，プログラミング課題の評価の支援 [1] や，教師の学習者への円滑なフィードバックの支援 [2]，プログラミング初心者のデータサイエンスの学習の支援 [3] を目的としたインタラクティブなシステムが開発されている．

プログラミングに関連する教育や研究が活発となっている一つの大きな要因として，情報産業界の人材の質および量の絶対的な不足が指摘されている．産業界における情報技術を習得した人材の重要性がますます増大する一方で，教育現場にて習得可能な情報技術と情報産業界で求められる能力には依然として大きな乖離が存在している．教育現場では情報技術の基本原則を習得することを目的としているが，情報産業界において求められる能力は，ソフトウェア開発プロジェクトを推進するための新機能開発やバグ修正といった実践的な技術である．そこで本研究では，以上のような教育現場と情報産業界の乖離を埋めるために，情報産業界において行われたソースコード変更からプログラミング課題を自動生成するシステムを提案する．

2　実現を目指すシステム

現実のソフトウェア開発において行われたソースコードの変更データを収集するために，本研究ではソフトウェア開発を管理するウェブ上のプラットフォームである GitHub[*1] のイシューとプルリクエストを使用する．GitHub が提供するイシューは，ソフトウェア開発における課題を登録し管理する機能である．イシューを導入することにより，新機能開発や修正すべきバグといった対応すべき課題を開発チーム内にて円滑に共有することが可能となる．イシューはタイトル・説明文・ラベルなどの情報から構成される．GitHub のプルリクエストと呼ばれる機能は，イシューを解決するためのコード変更を管理する機能である．プルリクエストはタイトルや説明文などのイシューと同様の情報に加えて，イシューを解決するためのコード変更の情報から構成される．

多くのソフトウェア開発者が GitHub を利用しており，2016 年 9 月から 1 年の間に約 1200 万件のイシューが作成された[*2]．従って GitHub には実践的なソースコード変更の膨大なデータが蓄積されており，それらから生成されるプログラミング課題はより実践的な情報技術の習得を支援することができると考えられる．本研究が提案するイシューとプルリクエストからプログラミング課題を生成するシステムの概念図を図1に示す．イシューの説明文は対応すべきソフトウェア開発の課題を説明し，その課題を解決するために必要なコード変更はプルリクエストに含まれている．プログラミング課題も同様に，解答すべき課題の説明と，その解答となるソースコードから構成される．従って，イシューとプルリクエストからプログラミング課題に適した情報を抽出することで，自動でプログラミング課題を生成することが出来ると考えている．

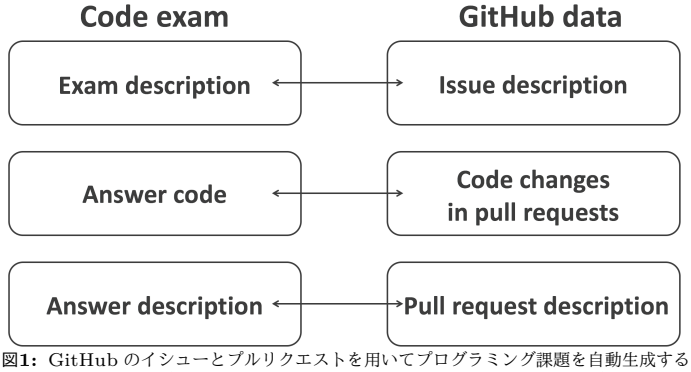

図1: GitHub のイシューとプルリクエストを用いてプログラミング課題を自動生成するシステムの概念図．プログラミング課題とその解答の説明文を，イシューとプルリクエストの説明文から生成する．さらに解答となるソースコードをプルリクエストのコード変更から生成する．

3　GitHub 上のコード変更の分析

GitHub 上の全てのイシュー及びプルリクエストを，プログラミング課題の自動生成に活用できるとは限らない．何故なら，説明が一切無いイシューや，あまりにも複雑すぎるコード変更を含むプルリクエストも多く存在するからである．プログラミング課題の作成者と解答者の両者の観点から考えると，プログラミング課題を自動生成する際に満たすべき条件を次のようにまとめることができる．

条件 1　プログラミング課題の説明文が，生成元であるリポジトリの管理者だけでなく，リポジトリを管理していない第三者から見ても理解可能であること．

条件 2　プログラミング課題が，リポジトリのファイル構造に関する知識を必要とするような大規模なコード変更量を要求していないこと．

本章では，上記の課題をどのように解決できるかを，収集したイシューとプルリクエストの分析を通して検証する．

3.1　データ収集

Up For Grabs[*3]というウェブサイトでは，フリーランスの開発者向けに，400 以上の GitHub 上のリポジトリと，それらのリポジトリ内において第三者であっても取り組み可能と指定されたイシューのラベルが掲載されている．これらのイシューはリポジトリの管理者ではないフリーランスの開発者向けに記述されているため条件 1 を満たす．そこで，Up For Grabs に掲載されたイシューと関連するプルリクエストから，条件 2 を満たすプログラミング課題を生成可能かどうか検証する分析を行った．

Up For Grabs に掲載されているイシュー及びプルリクエストのデータを GitHub から収集するために，GitHub の GraphQL API v4[*4]を使用した．実際に収集したイシューは約 43 万件，プルリクエストは約 25 万件である．

Examining the feasibility of automatic code exercise generation using GitHub pull requests

Daisuke SHIBATO† and Koji YATANI†

† Interactive Intelligent Systems Laboratory,
Graduate School of Engineering, The University of Tokyo
{shibato, koji}@iis-lab.org

*1 https://github.com

*2 https://octoverse.github.com/

*3 http://up-for-grabs.net

*4 https://developer.github.com/v4/

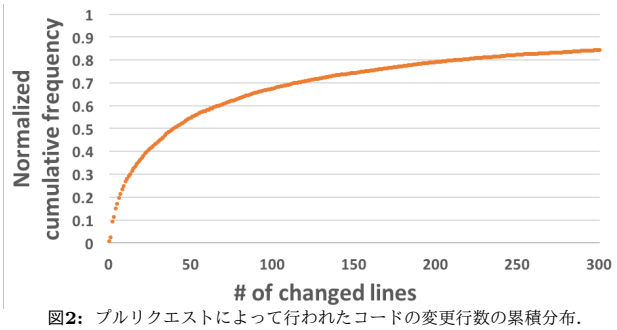

図2: プルリクエストによって行われたコードの変更行数の累積分布.

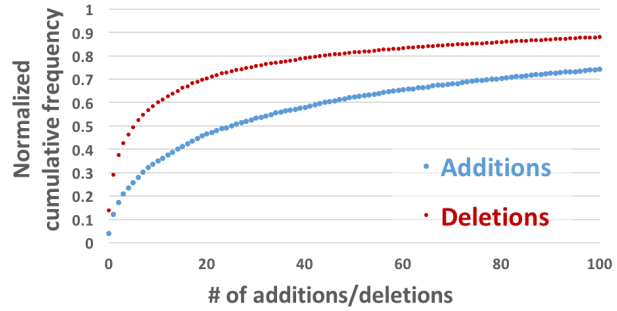

図3: プルリクエストによって行われたコードの追加および削除行数の累積分布.

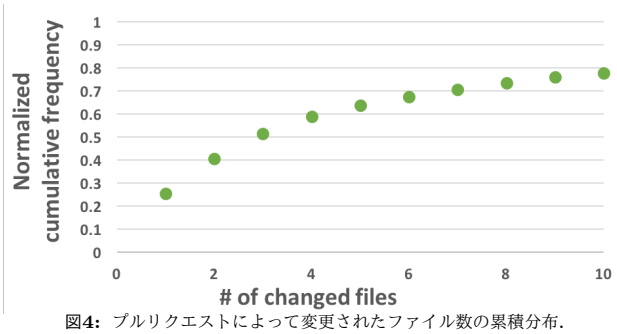

図4: プルリクエストによって変更されたファイル数の累積分布.

3.2 分析結果

コードの追加・削除行数

条件 2 で述べたように，自動生成されたプログラミング課題が，数百行に及ぶ膨大なコード変更を要求することは望ましくない．そこで条件 2 を検証するために，プルリクエストによって行われたコードの変更行数の分析を行った．プルリクエスト内にて変更された行数の累積分布を図2に，追加・削除された行数の累積分布を図3に示す．

図2から，40 行以下のコード変更から構成されるプルリクエストが全体の 50% 以上を占めることが分かる．また図3が示すように，プルリクエスト内のコード削除の 50% 以上が 10 行以下，コード追加の 50% 以上が 30 行以下であることが分かる．基本的な文法知識を問うプログラミング課題は，数十行のコード変更を必要とすることが一般的である．従って，GitHub のイシューとプルリクエストを活用することで，一般的にプログラミングの基礎レベルとされる課題が要求するコード変更量からなる，プログラミング課題を生成可能であることが分かる．

コードの追加・削除の組み合わせ

プルリクエストのコード変更には，新規コードの追加のみ，既存コードの削除のみ，及び両者を組み合わせた既存コードの変更の 3 種類がある．プルリクエストにおけるそれぞれのコード変更の頻度を分析した結果を表1に示す．既存コードの変更が 82.20% と最も多く，既存コードの削除のみが 3.89% と最も少ないことが分かる．

既存コードの削除や変更は，リポジトリの開発背景や構造に関する知識を必要とする恐れがある．そこで本研究では，新規コードの追加のみを行っているプルリクエストのみに焦点を当てることとする．

表1: プルリクエストにおける，新規コードの追加のみ・既存コードの削除のみ・両者の混合（既存コードの変更）の，それぞれのコード変更種類の頻度.

Category	Frequency
Only additions	13.91%
Only deletions	3.89%
Both	82.20%

変更ファイル数

条件 2 で述べたように，自動生成されたプログラミング課題が，複数のファイルに及ぶコード変更を要求することは望ましくない．何故なら，複数ファイルへのコード変更を行うためには，リポジトリのクラスやファイル構造に関する前提知識を必要とするからである．そこで，収集したプルリクエストにおける変更ファイル数の分析を行った．図4に，プルリクエストによって変更されたファイル数の分布を示す．

分析結果から，単一ファイルのみを変更したプルリクエストが全体の約 25% を占めていることが分かる．一方で，10 よりも多いファイル数を変更したプルリクエストが，全体の約 23% を占めている．複数のファイルに及ぶコード変更を要求するプログラミング課題は，解答者のリポジトリやそのファイル構造に関する知識や理解を要求する恐れがあるため，本システムの対象から除去する必要があると考えられる．

まとめ

前述の分析結果を踏まえて，基礎レベルのプログラミング課題が要求するコード変更を，単一ファイルに対する 50 行以下の新規コード追加のみと定義したとする．この時，筆者が作成したデータセットの内 1.6% のプルリクエストが条件を満たすことが分かった．GitHub 上のデータが膨大であることを考えれば，プログラミング課題を生成するために十分な割合と言える．

4 おわりに

本稿では，GitHub のイシューとプルリクエストを用いてプログラミング課題を自動生成するシステムの提案を行った．本研究の予備実験として，実際のイシューとプルリクエストのデータ分析から，3章の冒頭にて述べた条件を満たすプログラミング課題を生成可能であるか検証を行った．その結果，一般的な基礎レベルのプログラミング課題が要求するコード変更量と，同等の変更量からなるプルリクエストが実在することを明らかにした．今後は得られた知見をもとに，以下の課題に取り組む予定である．

- 膨大な GitHub のイシューとプルリクエストから，プログラミング課題に適した情報を抽出する機構の設計
- 抽出されたイシューとプルリクエストからプログラミング課題を生成・表示するシステムの設計

参考文献

[1] Andy, N., Christopher, P., Jonathan, H., and Leonidas, G.: Codewebs: Scalable Homework Search for Massive Open Online Programming Courses, Proc. *WWW '14*, pp.491–502, ACM (2014).

[2] Philip, J. G.: Codeopticon: Real-Time, One-To-Many Human Tutoring for Computer Programming, Proc. *UIST'15*, pp.599–608, ACM (2015).

[3] Xiong, Z. and Philip, J. G.: DS.Js: Turn Any Webpage into an Example-Centric Live Programming Environment for Learning Data Science, Proc. *UIST '17*, pp.691–702, ACM (2017).