

プログラム読解学習支援システムの開発と プログラムコードの困難度定量化

Developing a Program Reading Based Learning Support System and Quantifying Difficulty of Program Codes

杉本 稜*, 松本 慎平*¹, 加島 智子*², 山岸 秀一*¹
Ryo SUGIMOTO*¹, Shimpei MATSUMOTO*¹, Tomoko KASHIMA*² and Shuichi YAMAGISHI*¹

*1 広島工業大学 情報学部

*1 Faculty of Applied Information Science, Hiroshima Institute of Technology

Email: b212063@cc.it-hiroshima.ac.jp

*2 近畿大学 工学部

*2 Faculty of Engineering, Kindai University

あらまし: プログラミングの学習現場においては、プログラミングに抵抗なく取り組める層とそれを特に不得手とする層の極端な特徴を示す学習者層が従来から確認されており、プログラミングを不得手とする層を支援する仕組みが必要であると考えられる。本研究では、利用者の理解度に応じて問題を出題することができ、自学学習環境として利用可能なプログラミング読解学習支援システムを開発すること、これを用いてプログラムコードの困難度を項目反応理論により定量化することを目的とする。開発システムにより、学習者のプログラミング技能やプログラムコードの困難度を判定するためのデータを収集できるため、困難度の定量化が可能となる。実験の結果、プログラム読解が困難な書き方を確認し、プログラミング教材を構築する際や教授法を検討する際の参考となり得る知見を明らかにした。

キーワード: プログラミング, 学習支援システム, 読解学習, 困難度, 項目反応理論

1. はじめに

プログラミングの学習現場では、プログラミングに抵抗なく取り組める層とそれを特に不得手とする層の極端な特徴を示す学習者層が従来から確認されている。プログラミング教育を普及させるに際しては、プログラミングを不得手とする学習者の存在を無視することはできない。とりわけ、日本のソフトウェア開発力の向上は国際競争の中で不可欠であり、また、ソフトウェア開発に対する社会の要望は今後一層拡大することが予測される。したがって、プログラミングへの苦手意識を抑制させることが可能な教材の開発や、従来素養がないと判断されてきたプログラミングを不得手とする層を的確に支援する仕組みが必要であると考えられる。

このような背景のもと、プログラミングにおける技術要素に対する到達水準を明確なものとし、利用者の理解度に応じた成熟的学習活動を可能とする教材の開発を目標に研究が進められている⁽¹⁾。本研究では、利用者の理解度に応じて問題を出題することができ、自学学習環境として利用可能なプログラミング読解学習支援システムを開発すること、これを用いてプログラムコードの困難度を項目反応理論により推定することを目的とする。開発システムにより、学習者のプログラミング技能やプログラムコードの困難度を判定するためのデータの収集が容易となる。実験の結果、プログラム読解が困難な書き方を明らかにし、プログラミング教材を構築する際や教授法を検討する際に参考となり得る知見を得た。

2. 開発システム

本研究で開発するシステムは、C言語の読解学習を対象としたものであり、ソースコードを自動生成する。生成されるソースコードは特定の目的の処理を行うものではなく意味のない処理の羅列であるが、「プログラミングの素質はある無意味な規則を首尾一貫して適用できるかどうか依存している」との前提から、意味を持たないソースコードの読解であってもプログラミング読解の訓練には有効に働くのではないかと考えている。また、生成されるソースコードは数十行の短いものであるが、「1メソッドは1画面に収まる程度とする」という設計論は一般的であり開発現場で採用されることが多いことから、1ページ内に収まる処理の読解を行うことができる

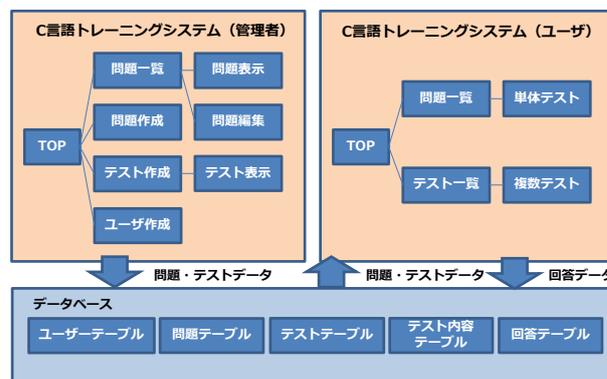


図1 システムの概要

よくなれば、大規模なプログラム読解も抵抗なく対応できるようになるのではないかと考えている。以上に基づきシステムを設計し開発した。

開発システムは Web ブラウザ上で利用する。システムは Apache 2.4.7 で動作しており、問題提示には JavaScript ライブラリである jquery 1.7.2、問題データ管理には MySQL 5.6.16、Web アプリケーションは php 5.5.9 を用いている。システムの概要を図 1 に、問題作成画面の動作画面を図 2 にそれぞれ示す。問題生成機能では任意の規則に基づいて問題を生成できる。テスト画面ではテストを作成することができる。画面上にソースコードと 6 択の選択肢が表示される(図 3 参照)。

3. 実験及び結果

C 言語プログラミングの基本を既に取得している大学 4 年生 10 名の被験者に対して、言語仕様を完全に理解していれば解ける知識問題 15 問、論理的な思考力が問われる論理問題 15 問の 2 つのテストをそれぞれ 10 分の制限時間で行い、回答ログデータの分析を行った。項目反応マトリクスを出力した後、4 母数ロジスティックモデル⁽³⁾により各問題のパラメータを計算した。

分析結果の一例として、ロジスティックモデルの上方漸近パラメータが他の問題と比較して有意に低かった問題を図 4 に示している。両者ともに、加算・除算代入やインクリメントが記述されており、これがプログラミングを得意とする学習であってもプログラム理解効率・可読性を妨げている原因であると考えられる。よって教授の現場では、これらの記述に対する指導を徹底することや、あるいはコーディング規約で禁止するなどの対応の必要性が示唆された。

4. おわりに

本研究では、プログラミング読解学習支援システムの開発と技能分析を行った。また、本システムを用いた分析結果の一例を示した。分析結果を踏まえて、コード記述の可読性が正答率に与える影響を分析する予定である。

謝辞

本研究は、独立行政法人日本学術振興会平成 25 年度科学研究費助成事業(若手(B) 13304922)及び平成 26 年度科学研究費助成事業(基盤研究(C) 26350296)の助成を受けて実施した成果の一部である。ここに記して謝意を表します。

参考文献

- (1) 北川 他, プログラム読解学習支援システムの開発と技能分析, 第 16 回 IEEE 広島支部学生シンポジウム論文集, B-43, In CD-ROM (2014).
- (2) 植野真臣, 荘島宏二郎, 学習評価の潮流, 朝倉書店 (2010)

```

戻る 変数数: 4 文数: 5 工数: 1 変数出現率: 100%
演算子: 無し 代入演算子: 代入 判定: 数値のみ
初期化あり インクリメントあり 無駄有り iあり whileあり
説明:
NEXT 保存

#include<stdio.h>
int main(void){
    int a,b,c,d;

    c = 3;
    while(c<3){
        b = 4;
        c++;
    }
    d = b;
    if(6>=9){
        c = d;
    }
    d = b;
    return 0;
}
    
```

図 2 問題作成画面

```

4:51 前の問題 3/4 次の問題 終了 戻る

#include<stdio.h>
int main(void){
    int a,b,c;

    c = 7;
    b = c / 2;
    c -= 2 + 5;
    b += c + 9;
    a = c + 5 - 10;
    return 0;
}

1: a=10 b=10 c=2
2: a=8 b=2 c=4
3: a=6 b=7 c=2
4: a=8 b=8 c=1
5: a=-5 b=12 c=0
6: a=7 b=7 c=1
    
```

図 3 テスト画面

```

#include<stdio.h>
main(void) {
    int a, b, c, d;

    b = 3 * 9;
    b += 1 - 5 * b;
    a = b - 7;
    b += 8;
}
        
```

```

#include<stdio.h>
main(void) {
    int a, b, c;

    a = 10 - 2;
    c = 7 - 2;
    while(a<2) {
        c /= 9;
        a++;
    }
    a += 2 * 4;
}
        
```

図 4 上方漸近パラメータが有意に低かった知識問題(左)と論理問題(右)