

プログラミング学習支援における問題自動生成に関する基礎的検討 Automatic Generation of Fill-in-the-blanks-type Questions for Programming Learning Support

野上 裕二[†] 納富 一宏[†]
Yuji Nogami Kazuhiro Notomi

1. はじめに

プログラミング入門者を対象としたプログラミング教育では、アルゴリズムの理解やプログラミング言語の文法の理解が必要不可欠であり、具体的なソースプログラムのコーディング方法やスタイルを学ぶことが実践的なプログラミングをマスターする上では重要となる。そのため、多くのサンプルプログラムを読解したり、プログラムの記述の仕方を真似たりすることが初期のプログラミング学習の基本となる。筆者らは、ソースプログラムから学習用の空欄補充問題^{[2]-[4]}を自動生成することに関して、主要なコーディングスタイルの獲得という点に着目した基礎的な検討とオンライン学習システムの開発を行っている^[1]。本稿では、初学者を対象としたプログラミング学習における空欄補充問題の自動生成手法の提案を行う。

2. 問題自動生成

2.1 ソースコードを基にした問題生成

任意のソースコードからプログラミング問題を自動生成するためには、ソースコードに含まれる、関数定義、変数宣言、制御構造などの統語情報を用いた要素をキーとして出題対象を定める。そして、そのキーを基にソースコードからブランクとフィルアを作成する。ここで、ブランクは出題時に設問となるデータ、フィルアは解答となるデータとなる。これらをペアにして、データベースに保存する。一連の流れを示したものを図 1 に示す。

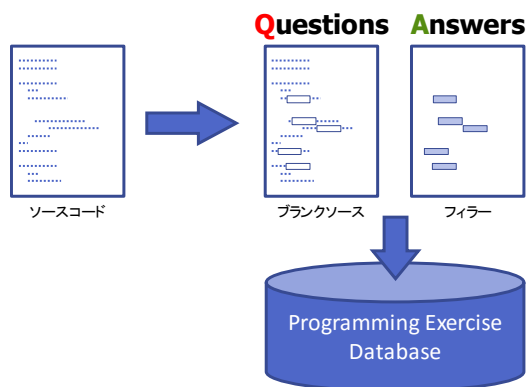


図 1 問題自動生成の流れ

2.2 ソースコードを基にした問題生成

自動生成される問題の方式として、ソースコードのタイピング練習を目的とした、(a)「プログラムを写す解答方式」や、予約語などの構文要素および頻出する単語への理解を深めるための(b)「制御構造、型、関数名をキーワードとし

た空欄方式」、コーディング要素の決定に際し、空欄方式よりも敷居を下げるための(c)「多肢選択方式」、さらに回答の自由度をあげ柔軟に対応する力を養う、(d)「コメントを基に問題を生成する方式」などが考えられる。図 2 にこれらの方式の実装イメージを示す。

0. 以下のプログラムを入力しなさい。

```
// 関数pyramidの定義
void pyramid(int h){
    int i;

    // 行毎に処理する
    for(i = 1; i <= h; i++){
        putchar(' ', h - i); // 空白の描画
        putchar('*', 2 * i - 1); // *の描画
        printf("\n"); // 1行の終わりの改行
    }
}

void pyramid(int h){
    int i;
```

(a) プログラムを写す方式の例

1. 以下の空欄を補充しなさい。

```
// 関数pyramidの定義
void _____(int h){
    int i;

    // 行毎に処理する
    _____(i = 1; i <= h; i++){
        _____(' ', h - i); // 空白の描画
        _____('*', 2 * i - 1); // *の描画
        printf(_____); // 1行の終わりの改行
    }
}

void _____(int h){
    int _____;
```

(b) 空欄方式の例

2. 以下の空欄に入れるべき適切な選択しをリストから選択しなさい。

```
// 関数pyramidの定義
void pyramid(_____)(int h){
    int i;

    // 行毎に処理する
    _____(i = 1; i <= h; i++){
        _____(' ', h - i); // 空白の描画
        _____('*', 2 * i - 1); // *の描画
        printf(_____); // 1行の終わりの改行
    }
}

void _____(int h){
    int _____;
```

(c) 多肢選択方式の例

3. 前後のコメントを参考にして以下の空欄を補充しなさい。

```
// 関数pyramidの定義
_____ (int h){
    int i;

    // 行毎に処理する
    for(i = 1; i <= h; _____){
        _____ // 空白の描画
        _____ // *の描画
        _____ // 1行の終わりの改行
    }
}

void _____(int h){
    int _____;
```

(d) コメントから空欄を自動生成する例

図 2 自動生成される問題の方式の例

[†] 神奈川工科大学 Kanagawa Institute of Technology

プログラミング問題を自動生成するシステムを開発する流れとして、初めに関数定義、型、変数宣言、制御構造、関数呼び出しなどの構文情報をターゲットとした。次に変数名、関数名などの識別子についてもターゲットとし、現在はコメント情報をターゲットとするシステムの開発を行っている。ソースプログラム中の各ターゲットの例を図 3 に示す。なお、コメントの記述スタイルはコーディングスタイルに依存するものであり、学習者のスキルに応じたものに対応させていく必要があると考える。

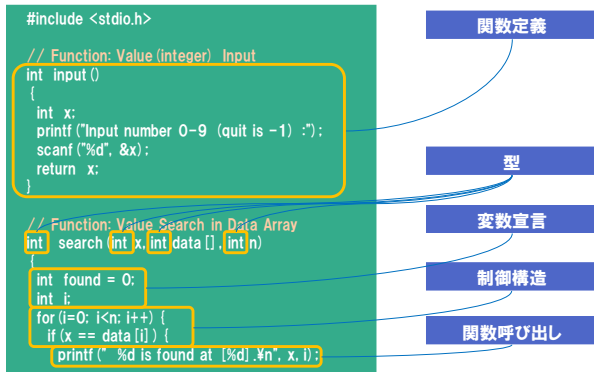


図 3 問題自動生成の処理対象の例

3. システム実装

3.1 システム構成

提案システムは HTTP サーバを経由して動作する、いわゆる Web アプリケーションとして構成されており、プログラムおよびデータはすべてサーバ側に存在する。ユーザはクライアントコンピュータからブラウザ経由でサーバに接続して利用することができる。システムはユーザインタフェースを介して各構成モジュールと相互に連携して処理を進める。具体的には、問題生成モジュール、予約語管理モジュール、解答管理モジュール、採点モジュールの合計 4 モジュールが存在し、これらは下位階層に存在する 2 つのデータ管理モジュールと連携する。システム構成を図 4 に示す。学習コンテンツはコンテンツ管理モジュールを介して入出力処理を行う。学習履歴は履歴管理モジュールを介して入出力処理を行う。

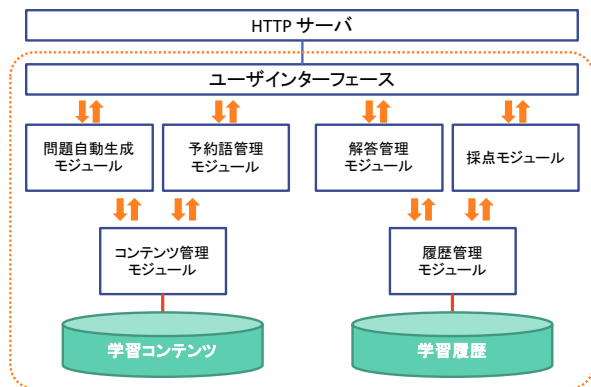
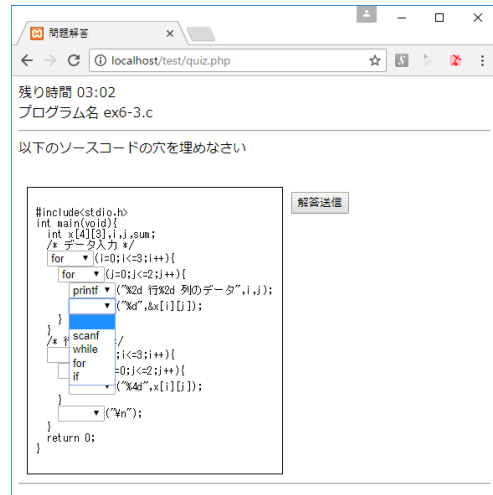


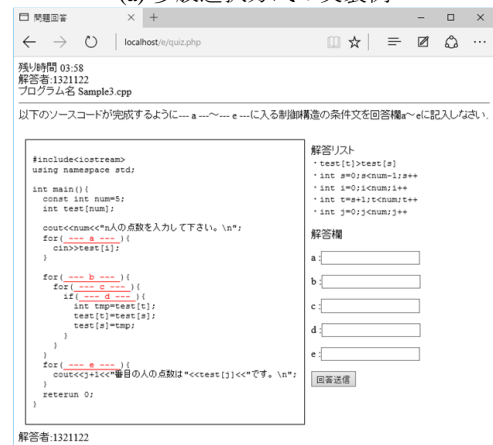
図 4 システム構成

3.2 システムの動作

システムの動作画面としてユーザによる解答画面の例を図 5 に示す。同図(a)は多肢選択問題の解答を行っている例で、(b)は制御構造の条件式を対象とした空欄方式である。



(a) 多肢選択方式の実装例



(b) 制御構造の条件式部分を対象とした空欄方式

図 5 動作画面例

4. おわりに

本研究では、プログラミング学習を支援するために任意のソースコードから問題自動生成を行うシステムの開発を進めている。現在、複数の問題方式の中で学習効果の高い方式を調べるための評価を行っている。

参考文献

- [1] 野上裕二：「プログラミング学習支援のための問題自動生成システムの開発」, 神奈川工科大学情報学部情報工学科平成 28 年度卒業論文, 神奈川工科大学, (2017)
- [2] 柳田峻, 他：「穴埋め問題を用いたプログラミング教育支援ツール pgtracer における教員用機能の実装」, 情報処理学会報告 Vol.2014-CE124 No. 6
- [3] 竹内 亮太郎, :「空欄補充問題の自動生成による Haskell プログラミング学習支援環境」, 情報処理学会報告 Vol. 2011-SE-171 No. 15
- [4] 有安浩平, 他：「学習者に合わせた C 言語演習穴埋め問題の自動生成」, 第 7 回日本データベース学会年次大会 DEIM Forum 2009 D9-5