

Java メソッドの自動生成を目指して

下仲 健斗^{1,a)} 肥後 芳樹^{1,b)} 楠本 真二^{1,c)}

概要：ソースコードを自動的に生成する、自動プログラミングと呼ばれる技術は古くから研究されている。これまで様々な手法が提案されてきているが、用途が限定的であったり、数行程度の短いソースコードしか生成できないものがほとんどである。本研究では、Java メソッドの仕様からソースコードを自動生成することを試みる。仕様は、シグネチャ情報（引数の型と返値の型）および入出力情報（引数の値と返値の組の集合）として、提案手法に与える。提案手法は、シグネチャ情報を用いて既存の Java ソースコードを探索し、生成する Java メソッドの基となりうるコードを発見する。そして、入出力情報を満たすように少しずつコードを進化させていく。本稿では、提案手法とその実装、および実際に生成した Java メソッドの事例を紹介する。

1. はじめに

自動プログラミングとは、開発者が作成したいソースコードを自動的に生成する技術のことである。つまり、開発者が直接ソースコードを書くのではなく、欲しいソースコードの挙動や仕様を断片的に与えると、それを満たすソースコードを出力する技術である。自動プログラミングという技術は古くから研究されており、これまで様々な手法が提案されてきているが、用途が限定的であったり、数行程度の短いソースコードしか生成できないものがほとんどである [1]。

本研究では、Java メソッドの仕様からソースコードを自動生成することを試みる。入力として与える Java メソッドの仕様は以下の 2 種類とする。

- シグネチャ情報（引数の型と返値の型）
- 入出力情報（引数の値と返値の組の集合）

提案手法の流れとしては、まず生成したい Java メソッドと引数と返値の型が一致したコード片を過去に作成された大量の Java メソッドの集合から探索する。その後、引数と返値の型が一致したメソッド群に対して、入出力情報を満たすように少しずつコードを進化させていく。提案手法を実装し、オープンソースソフトウェアを用いて手法を適用したところ、本手法が有用であるような事例を発見した。

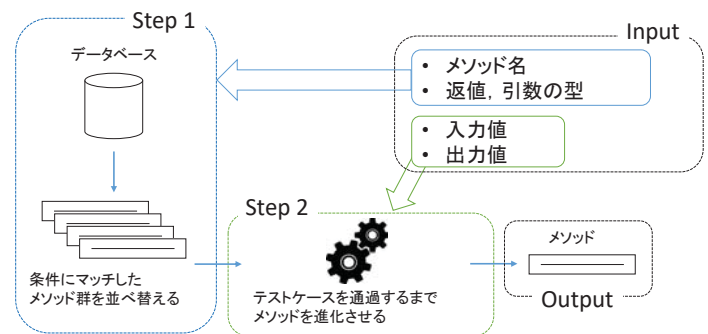


図 1 提案手法の概要

2. 提案手法

本研究では、Java メソッドの仕様からソースコードを自動生成する手法を提案する。提案手法の概要を図 1 に示す。提案手法の入力は Java メソッドのシグネチャ情報、および入出力情報である。出力は、仕様を満たす Java メソッドである。

提案手法は次の 2 ステップから構成される。

Step 1 メソッドの抽出

Step 2 メソッドの進化

Step 1 では、入力されたシグネチャと一致するメソッドをデータベースから抽出する。その後、抽出したメソッド群を優先度が高い順に並び替える。これは、抽出したメソッドの数が膨大なため、ランダムに選択すると効率が悪いためである。Step 2 では、入出力情報を満たすまで抽出したメソッドを進化させる。入出力情報を最初に満たした

¹ 大阪大学大学院情報科学研究科
Graduate School of Information Science and Technology,
Suita, Osaka 565-0871, Japan

a) s-kento@ist.osaka-u.ac.jp

b) higo@ist.osaka-u.ac.jp

c) kusumoto@ist.osaka-u.ac.jp

```
1 public boolean startsWith(final String str){
2     if (str == null){
3         return false;
4     }
5     final int len = str.length();
6     if (len == 0) {
7         return true;
8     }
9     if (len > (size)){
10        return false;
11    }
12    -int pos = size - len;
13    +int pos = 0;
14    for (int i = 0; i < len; i++ , pos++){
15        if ((buffer[pos]) != (str.charAt(i))){
16            return false;
17        }
18    }
19    return true;
20 }
```

図 2 提案手法によって得られたコード例

メソッドが最終的に出力される。

3. 実装

提案手法を実現するためには、以下の3点の実装が必要である。

- データベースの構築
- メソッドのランク付け
- メソッドの進化

本研究では Java を対象として提案手法を実装した。以降、3つの実装方法について述べる。また、apache-commons-text プロジェクト [2] を対象に、プロジェクト内にあるメソッドが自動生成できるかどうかを調査した。その結果についても述べる。

3.1 データベースの構築

データベースとして SQLite[3] を用いた。データベースに格納する属性は、返値と引数の型、メソッド名、ファイルパス、クラス名、プロジェクト名、メソッドの開始行、ソースコード、である。apache プロジェクトからランダムに5つのプロジェクトを収集し、約1万のメソッドを格納した。

3.2 メソッドのランク付け

メソッドのランク付けの指標として、メソッド名における類似度を用いた。そのため、提案手法の入力として、生成したいメソッドに対して適切なメソッド名を追加する。入力されたメソッド名と、抽出されたメソッド群との類似度をレーベンシュタイン距離を用いて算出する。その値が大きいほど、優先度が高いと判断する。

3.3 メソッドの進化

メソッドの進化は、遺伝的アルゴリズムを用いた。ツールとして、自動バグ修正ツールである GenProg[4] を用いた。GenProg は、同クラス内や同プロジェクト内に存在する文の挿入や置換、および文の削除を繰り返すことにより、メソッドを進化させる。メソッドの進化を入出力情報を満たすまで行い、入出力情報を満たしたメソッドを最終的に出力する。

3.4 実際に得られたコード例

apache-commons-text 内に存在する startsWith メソッドを、同プロジェクト内に存在する endsWith メソッドを進化させることにより自動生成することができた。図2に、提案手法を用いて得られた startsWith メソッドを示す。12行目の文が13行目の文に置換されていることが分かる。このように、処理内容が類似したメソッドを進化させることによって、自動生成が可能であることが分かった。

4. おわりに

本研究では、Java メソッドの仕様からソースコードを自動生成する手法を提案した。提案手法では、メソッドのシグネチャ情報と入出力情報を与えることで、目的のメソッドを得ることができる。

提案手法をツールとして実装し、オープンソースソフトウェアに対して適用した。その結果、処理内容が類似したメソッドを進化させることによって、自動生成が可能であることが分かった。

今後の取り組みとして、提案手法の評価実験を行ってみたいと考えている。実験の流れとしては、まずプロジェクトの開発履歴を解析する。そして、あるリビジョン r とリビジョン $r+1$ 間において、追加されたメソッドをテストデータとする。つまり、リビジョン $r+1$ に追加されたメソッドが、リビジョン r 内のメソッドを進化させることで自動生成できるかどうかを評価する。

謝辞 本研究は、日本学術振興会科学研究費補助金基盤研究 (S) (課題番号: JP25220003) の助成を得て行われた。

参考文献

- [1] 森畑明昌. プログラミングするプログラム-自動プログラム作成最前線. 情報処理, Vol. 57, No. 6, pp. 544-549, 2016.
- [2] Apache Commons Text. <https://commons.apache.org/proper/commons-text/>.
- [3] SQLite. <https://www.sqlite.org/>.
- [4] Claire Le Goues, ThanhVu Nguyen, Stephanie Forrest, and Westley Weimer. Genprog: A generic method for automatic software repair. *IEEE Transactions on Software Engineering*, Vol. 38, No. 1, pp. 54-72, 2012.