4F-2

可読性の指摘を行うプログラミング教育システムの開発 — 反復構造の自動検出による関数化の指摘 —

松本 章代 † Martin J. DÜRST ‡

東北学院大学 教養学部 † 青山学院大学 理工学部 ‡

1 はじめに

我々は，大学でプログラミング教育に携わっている。担当している演習授業では，提出された課題プログラムを自動チェックするシステムを構築し，数年前より運用している。これは，プログラムが正しく動くことを確認するもので，学生がプログラムをウェブブラウザ上からアップロードすると，システムがコンパイルおよび動作確認のテストを行う仕組みとなっている。現在は，このシステムに，可読性の指摘を行う機能の追加を検討している [1][2]。今回は，反復構造（コピー&ペーストを多用した類似処理）を自動検出することにより，関数化の指摘を行う機能を作成したので報告する。

2 研究背景

従来より，ソフトウェアの品質保証を目的とした静的コード解析ツールは，数多く存在している。これらの中には，IBM の Rational Logiscope [3] など，重複したコードを検出する機能が備わっている製品もある。しかしながら，これらのツールは大規模に開発される商用ソフトウェアを対象としているため，教育に利用するには不向きである。たとえば,「プログラマがポインタを未習得である」といった状況に対応した指摘は不可能である。

一方，教育目的で開発された静的コード解析ツールには，玉木らによる e ラーニングシステム [4] がある。これは，静的コード解析によって変数の初期化忘れや配列の範囲外アクセスなどの論理的な誤りを検出することを目的としており，可読性の低いコードを指摘するものではない。

これらの既存ツールに対し，我々のシステムは，授業において “Don’t Repeat Yourself” の考え方に基づいたプログラムの作成を身につけさせることを目的としている。よって,「関数にすべきコピー&ペースト」を検出するにあたり，ポインタ（引数の参照渡し）を教えた後かどうかを指定できるといった，授業内での使用に配慮した機能を持っている。

Development of an Education System Emphasizing Program Readability — Automatically Identifying Repeated Program Fragments for Conversion to Functions

†Akiyo MATSUMOTO, Faculty of Liberal Arts, Tohoku Gakuin University

‡Martin J. DÜRST, College of Science and Engineering, Aoyama Gakuin University

3 反復構造の検出手法

著者の松本は，ウェブページの中から反復構造を検出する研究 [5][6] に携わっている。そこで用いている配列アラインメントという手法を本システムでも利用し，ソースプログラムの中から関数にすべきコピー&ペースト，すなわち反復構造を抽出する。

3.1 配列アラインメント

配列アラインメントとは，複数の配列が入力されたときに文字間の対応関係を計算すること，もしくはその計算結果をいう。主にバイオインフォマティクスの分野でアミノ酸配列の解析などに用いられる技術であり，2 つの配列の類似性を適切に評価することができる。配列アラインメントは 2 個の配列に対するものと 3 個以上の配列に対するものとに大きく分けられ，2 個の配列に対する配列アラインメントを，特にペアワイズアラインメントという。本システムで利用するのはこのペアワイズアラインメントである。

本システムではまず，ソースプログラムを行単位でラベルに変換する。その際，一般に関数を作るときには変数や定数は抽象化できることに配慮する。ただし，変数をすべて同じとみなしてしまうと，アルゴリズムを正しく認識できない可能性が高い。そこで，定数および配列の添え字以外が一致している行に同じラベルを付けていく（図 1）。

そして，比較したい 2 つのラベル列を入力配列としてペアワイズアラインメントを計算し，そのスコアを類似度として扱う。スコアが閾値以上であれば，入力された 2 つのラベル列を同じものとみなす。

配列アラインメントを用いることにより，ラベル列が完全に一致していない場合にも，多少の差異を吸収し，柔軟な判断を行うことが可能となる。

3.2 スコア行列

配列アラインメントを計算する際に必要となるスコア行列を導出する手法について述べる。対象がアミノ酸の場合は，あるアミノ酸 a が別のアミノ酸 b に置換される確率を統計的に計算し，それをスコア化するという手法が一般的である [7]。しかしあらかじめすべての種類を用意しておくことができるアミノ酸とは違い，ラベルはソースファイルごとにまったく異なる。そのためすべてのソースコードに対応可能なスコア行列をあらかじめ用意しておくことはできず，各ソースごとにスコア行列を動的に生成する必要がある。

そこで本システムでは，スコア行列を生成するの

元のソースコード	変換後のソースコード	ラベル
total = 0;	total=_CONST_;	L16
for (i=0; i<num; i++) {	for(i=_CONST_;i<num;i++){	L10
total = total + Eng[i];	total=total+Eng[_INDEX_];	L23
}	}	L19
engAve = total / num;	engAve=total/num;	L24
total = 0;	total=_CONST_;	L16
for (i=0; i<num; i++) {	for(i=_CONST_;i<num;i++){	L10
total = total + Math[i];	total=total+Math[_INDEX_];	L25
}	}	L19
mathAve = total / num;	mathAve=total/num;	L26

図 1: ラベル付けの例

にラベル間の類似率を利用する。ソースコード中の定数・変数はすべて「V」に置換し，行頭から一致している文字の割合を計算する。たとえば，変換後のソースコードが `total=total+Eng[_INDEX_];` と `total=total+Math[_INDEX_];` であった場合，どちらも `V=V+V[V];` となり，類似率は 100%となる。

3.3 部分ラベル列の生成

ペアワイズアラインメントによって反復構造を判断するためには，1 つのソースファイルから生成したラベル列全体の中から，2 つの部分ラベル列を範囲が重ならないように抽出する必要がある。

部分ラベル列の決め方は，先行研究 [5] に準ずるが，C 言語ソースファイル中の反復構造の検出に対応できるよう，以下の変更を行っている。

- 先行研究では，反復構造は隣接していなければならなかったのに対し，本システムは隣接していない場合も検出する
- `break`，`continue`，`return`，`}` が，先頭行となっている部分ラベル列は候補から外す
- 3 行以上に限定する

3.4 反復構造の判定

ペアワイズアラインメントの計算結果によって「類似度が高い」と認められた 2 つの部分ラベル列は「反復構造候補」となる。1 つのソースファイルに対し，すべての「反復構造候補」が出そろったら，この中から最終的な「反復構造」を求め出力する。まず，一方の範囲が完全に同じ場合は，同じ「反復構造候補」とみなす。次に，範囲の一部が重なっている場合は，最大範囲のものを選択する。

3.5 参照渡しの必要性の判断

引数の参照渡しを学習済みかどうかで関数にできるかどうかの判断は変わる。本システムではこの指定が可能である。

参照渡しを使わなければ関数化できないかどうか判定するために，反復構造の候補箇所以外にも存在する変数に対し，候補範囲内で更新（=による代入や++など）される変数の数をチェックする。該当する変数が 1 つだけなら戻り値にすればよいが，2 つ以上ある場合は参照渡しにするしかない。

4 評価実験

本システムの出力結果に対し，第三者の目視による妥当性調査を行う。今回の調査では，学生が授業の課題として作成した C 言語のプログラム 1,057 本を対象とする。教育範囲の限定は行わないものとする。

ソースファイルごとに「関数やループにすべきコピー＆ペースト」の有無を目視とシステムでそれぞれ判断して，以下の A〜D の 4 つのグループに分類し，A については範囲の正解・不正解を付ける。

- A システム：有・目視：有
- B システム：無・目視：有
- C システム：有・目視：無
- D システム：無・目視：無

調査の結果，A が 503 本，B が 21 本，C が 105 本，D が 428 本となり，A の 503 本中，検出した範囲が正しかったのは 293 本（58.3%）であった。

5 まとめ

プログラミング教育において，可読性の高いソースコードを書くように指導することを目的とし，「関数化すべきコピー＆ペースト」を検出・指摘するシステムを作成した。今後は，検出精度の向上および教育成果の評価を目指す。

謝辞

本研究は文部科学省科学研究費補助金（基盤 C，課題番号 21500905）の交付を受けている。

参考文献

[1] 宮島 明寛，松本 章代，Martin J. Dürst：オープンソースを用いた C 言語記述スタイルの統計分析の試み，情報処理学会 第 71 回全国大会 (2009).

[2] 松本 翔太，松本 章代，Martin J. Dürst：C 言語用のプログラミングスタイル評価システムの構築，情報処理学会 第 72 回全国大会 (2010).

[3] Rational Logiscope，`http://www-01.ibm.com/software/awdtools/logiscope/`

[4] 大久保 和則，玉木 久夫：教育を目的とした C 言語ソース管理 API の作成とその応用，信学技報 ET，No.108，Vol.247，pp.17-22 (2008).

[5] 池田 彰吾，松本 章代，小西 達裕，高木 朗，小山 照夫，三宅 芳雄，伊東 幸宏：繰り返し構造を考慮した Web ページの見出しの階層構造の解析，情処研報 2008-DD-65，Vol.2008，No.34，pp.31-38 (2008).

[6] 沙 鵬，松本 章代，小西 達裕，高木 朗，小山 照夫，三宅 芳雄，伊東 幸宏：繰り返し構造の検出に基づく Web ページの見出しの階層構造の解析，情処研報，Vol.2010-DD-75，No.6，pp.1-8 (2010).

[7] 阿久津 達也：バイオインフォマティクスの数理とアルゴリズム，共立出版，pp.1-53 (2007).