

プログラミング演習のためのプログラムテスト支援機能

戸上稔崇*1・北英彦*1

Email: togami@ce.elec.mie-u.ac.jp

*1: 三重大学大学院工学研究科電気電子工学専攻

◎Key Words プログラムテスト, プログラミング演習, 演習システム

1. はじめに

情報化社会の発展に伴い、小中学校でもプログラミング教育を始めようとする動きがあるなど、その重要性が高まってきている。プログラミング能力向上に必要な条件として、間違いのない品質の良いプログラムを書くことができるかという点がある。しかし、プログラミング演習の授業ではソースコードを記述することに主眼が置かれ、プログラムの品質を高めるための教育は行われていない。

プログラムの品質を向上させる方法のひとつにフットウェアテストがある。本研究では、ソフトウェアテストの手法の一つであるホワイトボックステストを、本研究室が開発しているプログラミング学習支援システム PROPEL (PROgramming Practice Easy for Learners) ⁽¹⁾ と連動して行えるようにすることで、プログラミング初心者が自らの作成したプログラムを利用してソフトウェアテストの学習を行える環境を提案する。

2. ソフトウェアテスト

ソフトウェアテストは、プログラムの内部構造に注目し構造を理解した上で意図通りに動作しているかを確認するホワイトボックステストと、プログラムの入出力だけに注目しプログラムが仕様通り動作しているかを確認するブラックボックステストの 2 つに分けることができる。前者には、条件網羅、分岐網羅、命令網羅などの網羅基準がある。本研究では、学習者に自分のプログラムの動作の流れを確認させるために、分岐網羅テストを支援の対象とする。

3. これまでの試み

我々の研究グループは、高桑⁽²⁾がホワイトボックステスト、袁⁽³⁾がブラックボックステストについて、学習者がテストケースを作成する演習の環境を提案した。

ホワイトボックステストについては、分岐網羅を学習の対象とする。演習ではテストの対象とするプログラムを予め与え、学習者はプログラムの構造を読み解きながら分岐網羅率が 100%となるようにテストケースを考え実行する。

ブラックボックステストについては、境界値分析に基づくテストを学習の対象とする。演習ではテストの対象とするプログラムの要求を与え、学習者はそれを見て境界値となるテストケースを考え実行する。

以上の学習方法については、それぞれ学習効果を確認した。

4. プログラミング演習支援システム

我々は、プログラミング演習支援を目的としたシステムを開発・運用してきた。そのうちの 1 つが学習者のプログラミング作成状況の把握及び、作成の遅れている学習者への迅速な対処を目的とした、プログラミング演習システム PROPEL である。PROPEL のシステム構成を図 1 に示す。学習者は Web ブラウザ上でプログラムを作成することができる。講師は学習者のプログラム作成状況を素早く把握することができる。

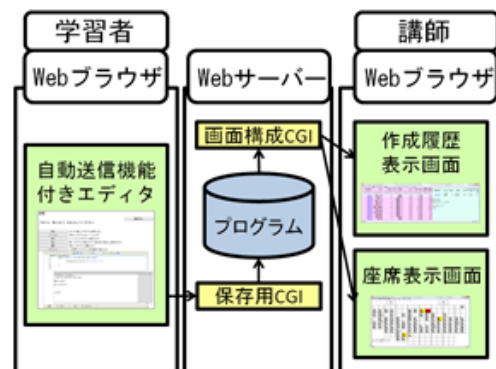


図1 PROPEL のシステム構成

コンパイルが成功してしまうと、学習者はプログラムが正しく動作するか十分に確認しないままプログラムを提出してしまう可能性がある。そういった学習者を把握するため、本システムには彦坂⁽⁴⁾が提案した自動テスト機能がある。講師が設定したテストケースを、提出されたプログラムに対して実行し、その結果が講師と学習者それぞれに提供される。彦坂の動作テストは提出したプログラムに対してシステムが動作テストを行うことを可能にする。しかし、システムがテストを行うのではなく、作成したプログラムに対して学生自身がテストケースを考え動作テストを行い、課題の要求通りに動作するかどうかを確認する必要がある。

5. プログラムテスト支援機能

提案するプログラムテスト環境を用いた学習のフローを図 2 に示す。学習者が PROPEL を利用して作成したプログラムからフローチャートを生成し、図 3 に示すプログラムテスト実行画面に移行する。学習者は、フローチャートとソースコードを見ながら、テストケースを考え入力する。入力されたテストケースをシステムが実行し、網羅率を測定する。以上を繰り返し、網羅率を高めていく。

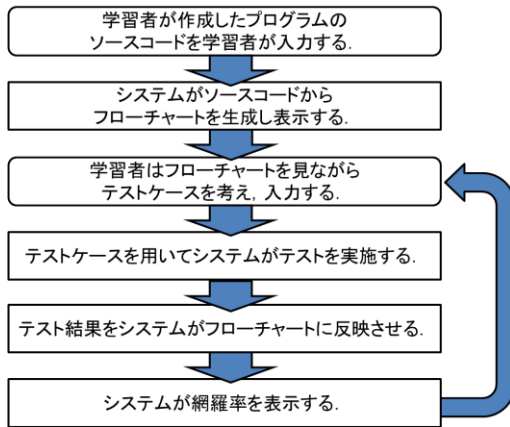
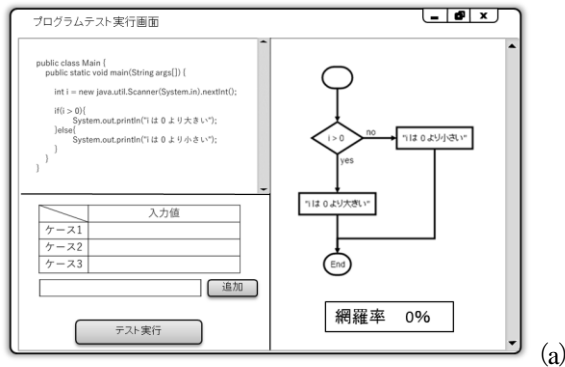
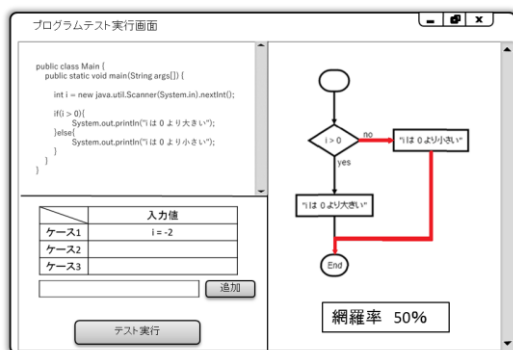


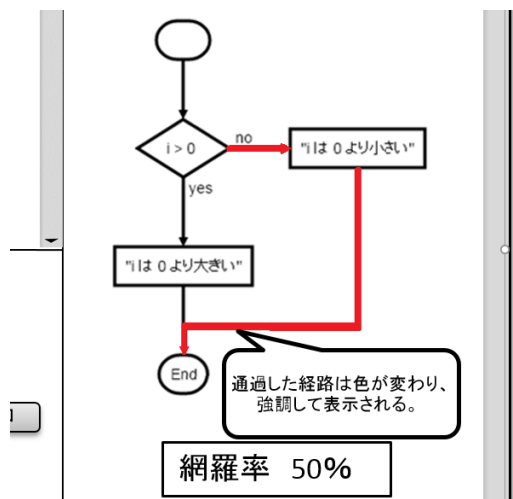
図2 学習フロー



テスト実行前



(b) テスト実行後

(c) フローチャート拡大図
図3 プログラム実行画面

学習者はプログラムを作成した後、コンパイルが成功したことを確認し図3 (a) の画面に移行することができる。画面左上には学習者が作成したプログラムが表示される。画面右には学習者がそこから生成されたフローチャートが表示される。学習者はそれらを見ながらプログラムの構造を視覚的に把握し、テストケースを考えることができる。テストケースは画面左下のフォームから入力し、リスト化される。考えるテストケースを作成し終え実行ボタンを押すと、プログラムが全てのテストケースを実行し、網羅率が計算される。網羅率は各分岐条件で真と偽がそれぞれテストされたかどうかで判断する。分岐が1つである図3の例であれば、分岐で真となる条件のみ実行されれば網羅率は50%である。テスト実行後の画面は図3 (b) のようになる。通過した経路はフローチャート上で強調して表示される。学習者は表示された網羅率とフローチャートを見て、網羅率が100%となるようにテストケースを修正、追加していく。

フローチャートの生成には flowchart.js というオープンソースツールを利用する。このツールは専用の文字列を入力することにより、その文字列に対応するフローチャートを生成する。学習者が作成したプログラムを、flowchart 専用文字列に変換することでフローチャート生成を可能にする。

6. 期待される効果

本システムにより、学習者は自らの作成したプログラムに対して、フローチャートを用いて視覚的に、ホワイトボックステストを実行することができる。テストケースを作成・実行し、網羅率を高めていくことでプログラムの内部構造への理解が深まり、より高品質なプログラムを書けるようになることが期待される。

7. まとめ

今回、学習者のプログラミング能力向上のため、学習者が自らの作成したプログラムに対しホワイトボックステストをフローチャートで視覚的にサポートしながら行える環境をプログラミング演習システム PROPEL に組み込んだ。自力でコンパイルエラーを取り除く力のある学習者が、講師の手を煩わせずにより高品質なプログラムを作成するための学習を行うことで、講師の負担軽減に繋がることを期待する。

参考文献

- (1) 伊富昌幸, 北英彦, 高瀬治彦, 林照峯: コーディング状況に応じたアドバイスを可能にするプログラミング演習システムに関する研究, コンピュータ利用教育協議会, 2016 PC カンファレンス (2016)
- (2) 高桑稔, 袁智朝, 北英彦: プログラミング能力向上を目的としたプログラムテストの学習システムに関する研究, CIEC 春季研究会 2015 (2015)
- (3) 袁智朝: プログラミング能力向上を目的としたソフトウェアテストの自己学習システムに関する研究, 修士論文 (2016)
- (4) 彦坂知行, 多人数でのプログラミング演習における学習者のコンパイルエラー状況の把握システム, 修士論文 (2016)