

3. 研究方法

研究方法を図3に示す。

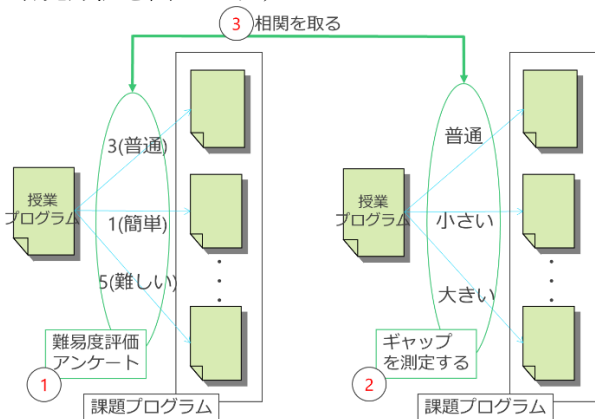


図3 研究方法

- ① 授業プログラム1つと課題プログラム幾つかをそれぞれ学生に提示し、授業プログラムに対し課題プログラムは簡単か難しいか、1~5の5段階で評価を行ってもらう。
- ② 評価してもらったものと同じ授業プログラム、課題プログラム間のギャップを、第2章で説明したメトリクスを用いて測定する。
- ③ 難易度評価アンケートの結果と、測定したギャップとの関係を分析する。

対象講義は、平成28年度奈良高専情報工学科2年生のプログラミングIである。前期講義15回のうち、5回の講義で課題プログラムに関するアンケートを実施した。各授業プログラムに対する課題プログラムの数を表1に示す。なお、講義を受講した学生は46人であったが、正しくアンケートに答えた学生は17名であった。

表1 課題プログラムの個数

講義	第5回	第6回	第7回	第9回	第10回
授業プログラム数	1	1	1	1	1
課題プログラム数	6	4	4	4	4

4. 研究結果

表1にプログラム間のギャップと学生の主観的な難易度との、相関係数とp値を示す。

表1 相関係数

用いたメトリクス	相関係数	p値
メジャーな構文要素数	0.680	0.009
マイナーな構文要素数	0.205	0.415
サイクロマチック数	0.313	0.156
行数	-0.099	0.660
ネスト数	-0.347	0.113
レーベンシュタイン距離	-0.562	0.007
ジャロ・ウィンクラー距離	-0.635	0.002

一般的に相関係数が±0.4~±1.0だと両者の間に相関があるとされる⁽⁴⁾ので、用いたメトリクスの中

で相関があるものはメジャーな構文要素数、レーベンシュタイン距離、ジャロ・ウィンクラー距離であることがわかる。

5. 考察

5.1 研究結果からの考察

メジャーな構文要素数は正の相関があったので、授業プログラム、課題プログラム間で、メジャーな構文要素数の差が大きければ大きいほど、学生が感じる難易度も上昇すると考えられる。また、レーベンシュタイン距離は負の相関があったので、授業プログラムから課題プログラムへの編集回数が多いほど、学生が感じる難易度も上昇すると考えられる。

5.2 今後の課題

本研究では、「授業プログラム」に対する「課題プログラム」の難易度評価を行ってもらったので、構文要素数をメトリクスとしてギャップを測定するとき、前回以前の講義プログラムを取り入れなかった。つまり、累積的な学習を前提としていなかった。しかし、実際学生は授業プログラムの構文要素のみを理解しているのではなく、前回以前の授業プログラムの構文要素も理解していると、考えられる。従って、前回以前の授業プログラムを考慮した分析方法の確立と実践が今後の課題となる。

6. まとめ

研究結果より、メジャーな構文要素、レーベンシュタイン距離、ジャロ・ウィンクラー距離をメトリクスとして選択し、定量的に計測されたプログラム間のギャップは、学生が感じる主観的な難易度と関係があることが分かった。つまり、これらのメトリクスを用いてプログラム間のギャップを縮めることで、教員はギャップの大きい課題を出題することが無くなり、学生は課題を解くのが困難ではなくなると考えられる。

謝辞

早く de-gapper を公開いただいた、明星大学情報学部情報学科長慎也准教授に感謝します。また、アンケートに協力いただいた学生の皆様に感謝します。

参考文献

- (1) 永野和男: プログラミングを通して学ぶ論理的な思考力と問題解決能力, CUBELAND[online], <http://www.cubeland.net/jirei_hm/500016/> (2005)
- (2) 長慎也, 保福やよい, 西田知博, 兼宗進「De-gapper-プログラミング初学者の段階的な理解を支援するツール」, 『情報処理学会論文誌』(55)No.1, pp.1-12(2013)
- (3) Daniel Jurafsky and James H. Martin. Speech and Language Processing (2Nd Edition), Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2009.
- (4) R+: 相関係数の意味と解釈[online], <http://rplus.wbnahe.info/statsemi_basic/sokankeisu.html>