

プログラミング演習における個別指導のための コーディング状況把握方法の提案 進捗度活動度平面によるサポート必要学習者の特定

2015SE044 久保田詩門

指導教員：蜂巢吉成

1 はじめに

大学におけるプログラミング演習では、学習者多数名、教員1名、TA少数名という形式で授業が行われる事が多い。講義スタイルの1つに、学習者が共通の演習問題に取り組み、教員とTA(以下、まとめて指導者と表記)が教室を巡回しつつ適宜指導を行うものがあるが、学習者の課題の進捗には個人差がある。また、どうしたら良いか分からずに数分間にわたり手が止まっている学習者もいる。しかし、演習時間内に指導者が教室内を巡回し全ての学習者の進捗を把握したり、どれだけ手が動いているのかを判断することは困難である。

本研究では五十里 [3] および川出 [4] が定義する進捗度を横軸、活動度を縦軸とした平面上に学習者の状況をグラフで表し、サポートが必要な学習者を特定する方法を提案する。

2 関連研究

井垣ら [2] は学習者のコーディング過程を分析し、可視化して教員へ提示するシステムを提案している。この研究ではコーディング過程やどのような誤りを行っているかを把握することは困難である。

石元ら [1] はソースコード中の演算と変数の型について同値類分割を行う方法を提案している。この研究は全体指導を目的としており、学習者個人の進捗を把握できるものではない。

3 進捗度活動度平面によるサポート必要学習者の特定方法の提案

本研究では、正解までの編集距離、コンパイル情報、実行情報をもとに評価する進捗度と、単位時間あたりに変化した文字数をもとに評価する活動度を利用する。これらの総合評価によって指導すべき学習者を個人レベルで把握する方法を提案する。

3.1 進捗度の定義

文献 [3] より、学習者の課題の進捗を数値化したものを進捗度とする。数値は0~100の範囲で、学習者のソースコードと模範解答との編集距離が小さいほど100に近く、編集距離が大きいほど0に近くなる。

3.2 活動度の定義

文献 [4] より、学習者がどのくらい作業しているかを表す活動度を用いる。活動度は0~100の範囲で、活動して

いるほど100に近くなる。

3.3 進捗度活動度平面および問別グラフ

横軸に進捗度、縦軸に活動度とした平面を進捗度活動度平面と定義する。進捗度活動度平面上で各学習者の各問を開始してから1分ごとに点をプロットして直線で結んだものを問別グラフとして定義する。問別グラフでは時間経過とともに点が左側(進捗度0)から右側(進捗度100)に移動していくと想定できる。

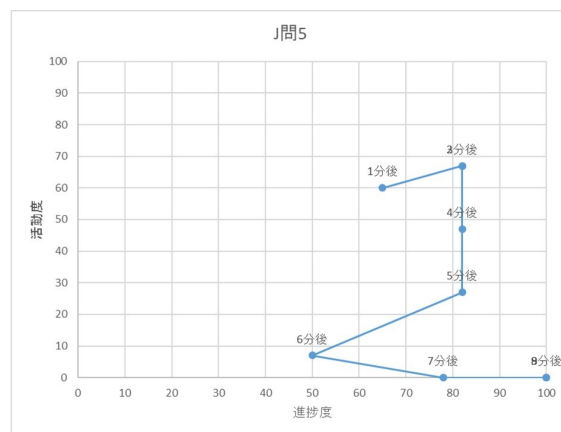


図1 問別グラフ：表示例

3.4 問別グラフによるサポート必要学習者の特定方法の提案

問別グラフを用いることで、進捗度と活動度の関係から学習者がどのように問題を解き進めているのかを把握することができる。

3.4.1 方法1

学習者が演習問題に解答するのにかかる平均時間を目安時間とする。目安時間は過去の類似の演習問題の経験や学習者の習熟度などから、あらかじめ指導者が設定するものとする。各問に取り組み始めて目安時間の半分が経過した時点で、課題の進捗を表す進捗度が半分である50以上である場合は順調に正解へと近づくことができているといえる。この視点から、目安時間の半分以上経過している時点で進捗度が50以下であり、かつ活動度が基準値である30を下回っている場合を想定する。このケースにある学習者は思うように正解に近付けておらず、なおかつ試行回数も減っており、指導が必要な可能性が高いと予測を立てた。

事象 1 その問を解き始めてから目安時間の半分以上経過している

事象 2 進捗度が 50 以下である

事象 3 活動度が基準値の 30 以下である

3.4.2 方法 2

進捗度が 80 以上である学習者は模範解答に近い状態まで進捗していると判断できる。しかしその状態で活動度が継続して下がっているということは、あと一步のところで行き詰まっていてサポートが必要だと予測を立てた。

事象 4 活動度が基準値 (30) 以下の状態が 2 分以上継続している

事象 5 進捗度が 80 以上である

4 実験

4.1 目的

3 節で提案した方法により、サポートが必要な学習者が特定できるか過去に行った演習データを用いて確認する。

4.2 方法

過去の演習データを用いて作成した問別グラフから、3 節の方法 1,2 に適合する学習者を抽出する。抽出した学習者が各問を解き終えるまでに要した時間を調べ、目安時間より長い、もしくは、解き終えていない場合をサポートが必要であったと判断する。演習データには、石元ら [1] の演習で得られたデータを用いる。C 言語を学んだ学部 3 年 11 名に協力してもらい、60 分の演習形式で、条件分岐、繰り返し、配列、ポインタ、再帰関数などの C 言語学習の主要な単元を網羅した問題を解答してもらった。60 分で計 5 題出題しているため、目安時間を 12 分とした。

4.3 結果

4.3.1 方法 1 について

11 人の学習者データから事象 1, 2, 3 を全て満たす部分は計 10 箇所検出された。このうち 7 箇所、つまり 70 % は解き終えていない問が含まれていた。計 10 箇所の問を解き終わるまでにかかった時間、もしくは解き終えていない問についてはその問を解いていた時間の平均は、目安時間の 12 分を越える約 18 分となった。上述の通りこの中には最後まで解き終えていない問が 7 箇所分含まれており、正答するまでに要する平均時間はさらに増えると考えられる。すなわち事象 1, 2, 3 を満たした学習者はサポートが必要な可能性が高いことが分かった。

4.3.2 方法 2 について

学習者 11 人分のデータから事象 4, 5 を共に満たす箇所は計 16 箇所検出された。このうちその箇所が含まれる問を解き終えるまでに要した平均時間を求めると、目安時間を越える約 14 分となった。ただし、この算出には最終的に解き終えていない問に取り組んでいた時間も含まれて

いるので、実際に解き終えるまでには更に時間がかかると考えられる。事象 4 かつ事象 5 を満たす区間で、最終的にその問を解き終えていない箇所は 12 箇所となり、約 75 % が解き終えていないことが分かった。これにより本研究で扱ったデータからは、事象 4 かつ事象 5 を満たした学習者はサポートが必要な可能性が高いと判断できることが分かった。

5 考察

今回の実験により、進捗度と活動度の総合評価によってサポートが必要な学習者を発見することができた。実際に演習を行いながら、本研究の提案方法を用いてサポートが必要な学習者を指導者へ提示する方法を考察する。学習者のリストアップをし、本研究の方法 1,2 に適合する学習者を色付けして表示することで、サポートが必要な学習者を視覚的に把握することができる。加えて本研究で扱った問別グラフも、学習者がどのようにその問を解き進めているのか判断することができるので、サポートを必要としている学習者の特定に有効であると考えられる。今後の課題としては、実際の演習中にどのようにサポートが必要な学習者を把握するのかという点が残っている。また、本研究ではサポートが必要な学習者の評価方法を提案したが、プログラミングの上達には試行錯誤する時間も必要であるため、本研究の提案方法を用いた場合の指導方法についても検討が必要になる。

6 おわりに

本研究では、学習者の個別指導のために学習者が記述したソースコードの中身に注目し、そこから進捗度と活動度を総合評価して提案方法の有効性を実験で確認した。今後の課題としては、学習者の記述傾向に沿った分析方法の実現、指導者の意向に合わせた他の設問や他の学習者を対象とした実験などによる検証が必要である。

参考文献

- [1] 石元慎太郎, 蜂巢吉成, 吉田敦, 桑原寛明, 阿草清滋: プログラミング演習における構文要素の種類毎のビューによるコーディング状況把握方法の提案, 情報処理学会, 情報教育シンポジウム, 8pages (2018) .
- [2] 井垣宏, 井上亮文, 齊藤俊, 山田誠: プログラミング演習における学生のコーディング過程可視化システム C3PV の提案, 情報処理学会論文誌, Vol.54, No1, pp.330-339 (2014) .
- [3] 五十里貴斗: 『プログラミング演習における個別指導のためのコーディング状況把握方法の提案-進捗度の観点から-』. 南山大学卒業論文, (2018)
- [4] 川出涼雅: 『プログラミング演習における個別指導のためのコーディング状況把握方法の提案-活動度の観点から-』. 南山大学卒業論文, (2018)