Updating STACK Potential Response Trees Based on Separated Concerns

https://doi.org/10.3991/ijet.v17i23.35929

Maryam Alarfaj[1,2](✉), Chris Sangwin[2]
[1]Saudi Electronic University, Riyadh, Saudi Arabia
[2]University of Edinburgh, Edinburgh, UK
m.alarfaj@seu.edu.sa

Abstract—The STACK system is a computer aided assessment package for mathematics. STACK questions include full algebraic input with validation. Feedback is provided to students using "potential response trees", written by the teacher for each question. In this research, we are updating STACK potential response tree based on students' concerns in learning induction proofs. We have identified a specific misconception in learning mathematical induction from prior educational research, designed STACK question to test if students exhibit this misconception, and then illustrated how we updated STACK potential response tree based on this misconception. One main goal of separating concerns is to understand students' responses and evaluate the question itself from an academic prospective and ultimately to improve the questions for future years. In addition, one of the contributions of this study is to improve our general understanding of how to design and use STACK potential response trees.

Keywords—STACK, mathematics, assessment

1 Introduction

STACK, a System for Teaching and Assessment using a Computer algebra Kernel, is an open source Computer Aided Assessment (CAA) system for mathematics, and other STEM subjects. The first version of STACK was developed in 2004 by Chris Sangwin in collaboration with Laura Naismith at the University of Birmingham. Since its first release, STACK has been continuously developed and is in widespread use particularly in higher education, notably by The University of Edinburgh, The Open University and Loughborough University. STACK focuses on accepting algebraic input from students. STACK grades students' answers through algorithms knows as "potential response trees" (PRT), where teachers can give partial marks and tailored feedback depending on the different mathematical properties of the students' answers. Recognition of mistakes through potential response trees, and the ability to have questions that require numerical or algebraic input for answers made STACK more powerful than the alternatives

In this research, we are updating STACK potential response trees in questions written to support learning mathematical induction. We have identified concerns in learning mathematical induction from prior educational research, designed STACK questions to test if students exhibit each misconception, and then illustrated how we updated STACK potential response trees based on these misconceptions. The main advantage of this approach is that we better prepare students to attend class by using online materials. In addition, one of the contributions of this study is to improve our general understanding of how to design and use STACK potential response trees.

2 Motivation

Online learning was the primary method to keep the learning process going during the Covid-19 period, which requires much support [1]. Students are increasingly being expected to use online assessment systems as support for traditional courses [2]. One possible approach to assessing problem solving has been to break up larger tasks into smaller individual questions to which e-assessment can then be applied. One drawback of breaking up questions is that it requires a significant investment of time and experience to develop a suitable question bank [3]. According to [4], automatically assessing a genuinely open-ended questions poses two main difficulties. First, the student must have a means to input their answer in a machine-readable format, which itself is nontrivial. Second, automation requires “preemptive” [2] decisions: the task designer must anticipate likely approaches from students and decide how they will be marked, before students have completed the task. Progress on this question could perhaps be achieved through cycles of design research, to develop prototypes of tasks, test them with students, and iteratively make improvements. The developers at the University of Edinburgh wanted to bring this concept to online assessment: to build an online assessment system that can carry out instructional scaffolding to help students to be prepared for class. STACK was ideal for this, as its features in providing specific feedback depending on the properties of students’ answers provide a great foundation for an adaptive system. The underlying goal of this research was to develop online assessments by transforming existing (largely paper-based) problem sets into online assessments.

3 Research

Separated concerns is a phrase used to describe materials in which potential misconceptions are addressed directly. By separating concerns, a specific issue is explicitly identified and addressed in advance of using it in a more substantial application. This is different from testing the students’ knowledge of a topic for its own sake. The idea of the separated concerns was heavily influenced by educational research. A number of concerns and misconceptions in learning mathematical induction were identified based on prior research and teaching experience discussed in [5]. For example, one of the common misconceptions in learning mathematical induction was dealing with sigma notation. So, by separating concerns, we design a STACK question involving the use of sigma notation with the expectation that a student will become generally more familiar and confident before they are asked to use this notation in an induction proof. One main goal of separating concerns is to understand students’ responses and evaluate the question itself from an academic prospective and ultimately to improve the questions for future years.

There are cognitive aspects linked to separating concerns, which we believe are potentially beneficial. The constraints in the interface might help students to be prepared for classes. Separating concerns might help to reduce the cognitive load where students can achieve success on the individual components. Separating concerns create a systemic environment in which misconceptions may be easier to spot before learning more complex materials.

3.1 Example 1

The following example for mathematical induction used to illustrates the idea of designing STACK question based on separating concerns.

Let $P(n)$ be the statement

$$\sum_{k=1}^{n}(2k-1)^2=\frac{n\cdot(2n-1)\cdot(2n+1)}{3}$$

1) Write the statement $P(n + 1)$
2) Calculate

$$\sum_{k=1}^{n+1}(2k-1)^2-\sum_{k=1}^{n}(2k-1)^2$$

Writing your answer in simplified form.

3) Calculate

$$\frac{(n+1)\cdot(2(n+1)-1)\cdot(2(n+1)+1)}{3}-\frac{n\cdot(2n-1)\cdot(2n+1)}{3}$$

There are three items in this example, where each item addresses one of the concerns in learning mathematical induction. In particular, item (1) addresses understanding what the induction statement $P(n + 1)$ is. Item (2) is designed to address dealing with sigma notation and how students manipulate an expression where all the action happen in the superscript and subscripts. Item (3) is designed to address some basic algebraic manipulation. It asks students to calculate the difference between the two fractions. The goal of item (3) is to help students to recognize that the first fraction is simply the RHS of $P(n + 1)$ and the second fraction is the RHS of $P(n)$ and so to encourage students to think before doing the calculations long-hand.

4 Development of STACK separating concerns tasks

Separating concerns tasks follow the scaffolding principle. A screenshot of STACK equivalent question of Example 1 is shown in Figure 1. Notice that the question contains three boxes into which the students should enter their answer, which we refer to as “interaction elements”.

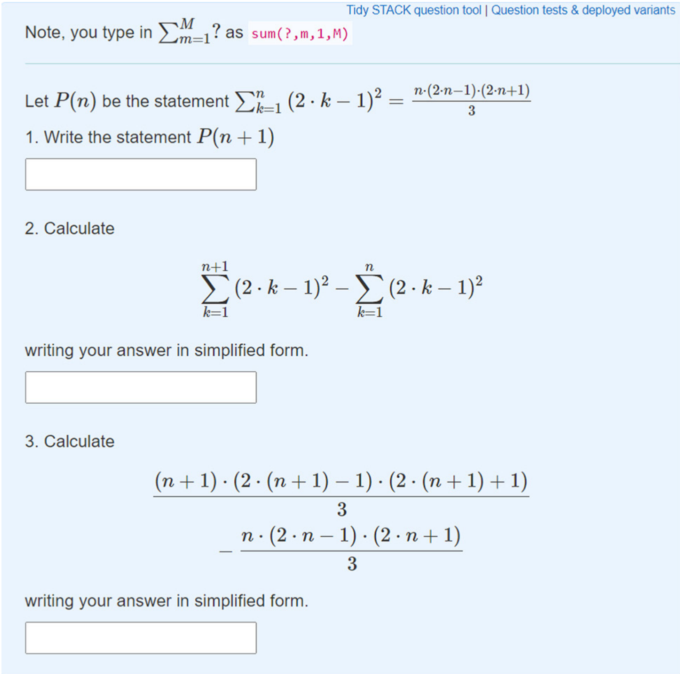

Fig. 1. STACK equivalent question of Example 1

Figure 2 illustrates the same web page, but with the student's answer typed in. A core part of the design of STACK is that students should type in a mathematical expressions as their answer.

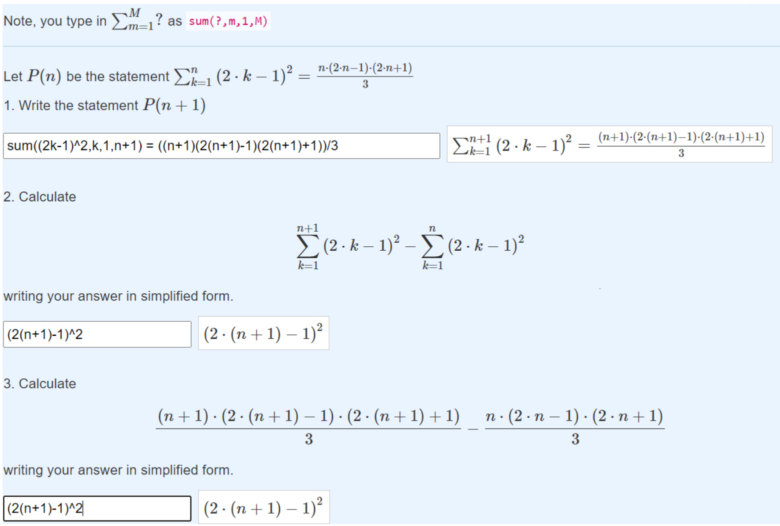

Fig. 2. STACK equivalent question of Example 1 with the student's answer

Notice that while simple algebraic expressions are not particularly problematic to type in, the sum requires knowledge of specific syntax. In particular, the student has to type in *sum* ((2*k* − 1) ^ 2, *k*, 1, *n* + 1) to represent $\sum_{k=1}^{n+1}(2k-1)^2$. Because such linear syntax differs between systems [6], and because students are known to find this so problematic, specific instructions have been provided at the top of the question, as can be seen at the top of Figures 1 and 2.

In this paper, to illustrate the idea of updating STACK potential tree response based on separating concerns, I only consider item (1) of Example 1 which addresses writing $P(n + 1)$. This concern was raised as an assessed question in 2020–21 in Proof and Problem Solving (PPS) course. We found that about 57.5% of students failed to type in $P(n + 1)$ as an equation. By reviewing students responses, we found that students either typed the LHS or the RHS of $P(n + 1)$.

At the outset, based on our prior experience as a teacher, we implemented a marking algorithm based on establishing that the students typed in $P(n + 1)$ as a correct equation. In particular, the following tests applied.

1. Is a student's left hand side (LHS) equivalent with the LHS up to commutativity and associativity of the elementary operations?
2. Is a student's right hand side (RHS) algebraically equivalent with the teacher's RHS?

Equivalence up to commutativity and associativity of the elementary operations is rather a subtle test, which is considerably stronger than full algebraic equivalence. In particular, we wanted to accept only answers which contain the sum operator $\sum$ but would condone both

$$\sum_{k=1}^{n+1}(2.k-1)^2 \text{ and } \sum_{k=1}^{1+n}(2.k-1)^2$$

But we would not want to accept other, algebraically equivalent, forms such as

$$\sum_{k=0}^{n}(2.k+1)^2,\ \text{or } \sum_{k=1}^{n+1}4k^2-4k+1$$

It is very unlikely that a student would type this, but nevertheless, we certainly don't want to accept the right hand side of $P(n + 1)$ which is algebraically equivalent to the left hand side as a correct form of the left hand side.

For the right hand side, we accept any algebraically equivalent expression.

The most direct answer would be to replace n with n+1 leading to

$$\frac{(n+1)\cdot(2(n+1)-1)\cdot(2(n+1)+1)}{3}$$

Recall that students must provide an answer which consists of a mathematical expression, and when valid, the system then seeks to establish properties of this expression. In STACK the "potential response tree" is the algorithm which establishes the mathematical properties of the student's answer and assigns outcomes. A potential response tree consists of an arbitrary number of linked nodes which are called potential response nodes. In each node two expressions are compared using a specified answer test, and the result is either true or false. The outcome of this answer test determines what happens next. Each true/false branch of the potential response node can (i) assign/modify the numerical mark, (ii) add feedback for the student, (iii) record an "answer note" and (iv) determine whether any further potential response nodes should be executed next or the assessment process should stop at this point. The directed graph is acyclic preventing an infinite assessment algorithm loop.

An illustration of the potential response tree for item (1) is shown in Figure 3. Note that this potential response tree consists of two nodes. Node 1 is to verify the (LHS) and Node 2 to verify (RHS) of students' input. Both the RHS and LHS are checked independently, so both nodes are always executed in this question.

Node 1: to test the whether the LHS is equivalent with the teacher's LHS up to commutativity and associativity of the elementary operations:

If true (prt1-1-T): the LHS is correct (0.5 mark)

If false (prt1-1-F): the LHS is wrong, feedback will be provided.

Node 2: to test the whether the RHS is equivalent with the teacher's RHS up to algebraic equivalence:

If true (prt1-2-T): the RHS is correct (0.5 mark)

If false (prt1-2-F): the RHS is wrong, feedback will be provided.

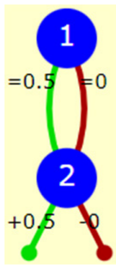

Fig. 3. The original STACK potential response tree of item (1)

Note, the immediate validity feedback in this question required an equation based on the original potential response tree in Figure 3. The original response tree has done a reasonable job in establishing the propose what students actually do, the response tree can be improved accordingly.

The updated potential response tree for item (1) is shown in Figure 4. This now has five nodes and is significantly more complex.

Node 1: to test whether or not the student's answer is an equation. On the basis of this test, two different situations are branched.

If true (prt1-1-T): the answer is equivalent to an equation. We proceed exactly as before; i.e. Node 2 tests the LHS and Node 3 tests the RHS. The above test is performed, and I have chosen to award half marks for each test.

If false (prt-1-1-F): the student's answer is non-equation. Then we proceed in a similar, but subtly different way.

Node 4: to test the LHS of the student's answer with the LHS of the teacher's up to commutativity and associativity.

If true (prt1-1-T): the LHS is correct (0.3 mark), and if so stop

If false (prt1-1-F): the LHS is wrong, feedback will be provided.

Node 5: to test the whether the RHS is equivalent with the teacher's RHS up to algebraic equivalence:

If true (prt1-2-T): the RHS is correct (0.4 mark)

If false (prt1-2-F): the RHS is wrong, feedback will be provided.

If the student's answer is equivalent to the LHS of the teacher's answer (Node 4) then it will also be algebraically equivalent to the RHS, which is why we stop when Node 4 returns true. We have chosen to award lower marks for each of these situations.

Note that for illustrative purposes here different marks result for each of the four end-nodes of the tree.

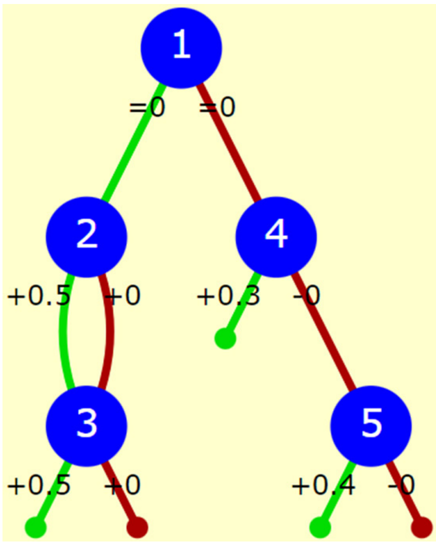

Fig. 4. The updated STACK potential response tree of item (1)

In fact, a slightly more useful way to look at the data for this question is to split up the answer notes and consider each node individually. This data is shown in Figure 5 together with brief narrative added in parentheses for the convenience of the reader.

```
## prt1 (426)
31  (  7.28%); !
137 ( 32.16%); prt1-1-F (Did not type in an equation)
258 ( 60.56%); prt1-1-T
42  (  9.86%); prt1-2-F
216 ( 50.70%); prt1-2-T (Correct LHS)
21  (  4.93%); prt1-3-F
237 ( 55.63%); prt1-3-T (Correct RHS)
88  ( 20.66%); prt1-4-F
49  ( 11.50%); prt1-4-T (Answer LHS only)
42  (  9.86%); prt1-5-F
46  ( 10.80%); prt1-5-T (Answer equivalent to RHS only)
```

Fig. 5. Split notes for the updated item (1) in STACK, with narrative

Node 1 checks if the student has entered an equation. Entering an equation is tested, in the revised tree, by prt1-1-T, which was given by 258 (60.56%) of the responses.

Surprisingly, 49 students appear to have typed in the LHS only, which is slightly more than the 46 who typed in the RHS only. Since both of these outcomes were generated for more than 10 % of the cohort it was well worth adding nodes to the tree with corresponding outcomes (e.g. feedback). It is probably not worthwhile adding nodes to test for strange individual responses.

5 Discussion and conclusion

This work has been carried for two terms 2020–21 and 2021–22 from the PPS course. The results reported in this paper consist of updated potential response trees to an example STACK question assessing mathematical induction. Note, this serves more generally as an example of how online assessment materials are designed and improved. It was also important that the course organizers were fully involved in the authoring process. They had to give clear and explicit guidance on the learning objectives and help review the mathematical content.

In general, we have found it much more sensible to operate a two year (at least) cycle of question development. First year cohort (2020–21): get the question working, with the essential properties for correct/incorrect established reliably. Second year cohort (2021–22): review students' data and update response trees accordingly, making sure the second (and subsequent) years benefit from better feedback (formative) or more nuanced partial credit (summative). In assessed quizzes, or online exams, marks are not available immediately. So we could improve the PRT before students marks are released to students. However, writing response trees is a significant amount of work. Experience suggests that a lot of time can be expended writing potential tests with feedback which are never actually used by any significant proportion of the students. Time is then wasted trying to second-guess what students might do. Even then, a review of students' responses is necessary because students write answers we might not actually anticipate. Here, we did not anticipate students typing the LHS, and the original tree probably would not have tested for this anyway. Reviewing students' answers closes the learning cycle for teachers by allowing them to understand what students are doing.

Creating online assessments is significant additional work, but once the questions have been created they require minimal work to maintain and can last for the lifetime of the course.

6 References

[1] H. R. P. Negara, Wahyudin, E. Nurlaelah, and T. Herman, "Improving Students' Mathematical Reasoning Abilities Through Social Cognitive Learning Using GeoGebra", *Int. J. Emerg. Technol. Learn.*, vol. 17, no. 18, pp. 118–135, 2022. https://doi.org/10.3991/ijet.v17i18.32151

[2] C. J. Sangwin, *Computer Aided Assessment of Mathematics*. Oxford, UK: Oxford University Press, 2013. https://doi.org/10.1093/acprof:oso/9780199660353.001.0001

[3] J. Anderson, A. Corbett, K. Koedinger, and R. Pelletier, "Cognitive Tutors: Lessons learned," *Journal of the Learning Sciences*, vol. 4, no. 2, pp. 167–207, 1995. https://doi.org/10.1207/s15327809jls0402_2

[4] G. Kinnear, I. Jones, C. Sangwin, M. Alarfaj, B. Davies, S. Fearn, C. Foster, A. Heck, K. Henderson, T. Hunt, P. Iannone, I. Kontorovich, N. Larson, T. Lowe, J. Meyer, A. O'Shea, P. Rowlett, I. Sikurajapathi, and T. Wong, "A Collaboratively-Derived Research Agenda for E-assessment in Undergraduate Mathematics," *International Journal of Research in Undergraduate Mathematics Education*, 2022. https://doi.org/10.1007/s40753-022-00189-6

[5] M. Alarfaj and C. J. Sangwin, "Mathematical Induction in Advanced Higher Mathematics," *Scottish Mathematics Council Journal*, vol. 50, pp. 79–86, 2020.

[6] C. J. Sangwin, and P. Ramsden, "Linear Syntax for Communicating Elementary Mathematics," *Journal of Symbolic Computation,* vol. 42, no. 9, pp. 902–934, 2007. https://doi.org/10.1016/j.jsc.2007.07.002

7 Authors

Maryam Alarfaj received her master's degree in applied mathematics from Oregon State University. She is now a lecturer in mathematics at the Saudi Electronic University and doing her Ph.D. at the University of Edinburgh. In 2022, she was awarded the status of *Associate Fellow* of the Higher Education Academy, UK (AFHEA). Her research interests lie in assessing proof construction and automatic assessment of mathematics using computer algebra (email: m.alarfaj@seu.edu.sa).

Chris Sangwin joined the University of Edinburgh in 2015 as Professor of Technology Enhanced Science Education. His learning and teaching interests include (i) automatic assessment of mathematics using computer algebra, in particular the development of the STACK system, and (ii) problem solving using student-centred approaches. In 2006, he was awarded a National Teaching Fellowship. Chris Sangwin is the author of a number of books, including *Computer Aided Assessment of Mathematics* and the popular science book *How Round is Your Circle*, which illustrates and investigates many links between mathematics and engineering using physical models. He has also co-authored a number of school textbooks published by Haese Mathematics (email: C.J.Sangwin@ed.ac.uk).

Article submitted 2022-09-10. Resubmitted 2022-10-19. Final acceptance 2022-10-24. Final version published as submitted by the authors.