A Learning Environment for Promoting Structured Algebraic Thinking in Children

Sergio Gutiérrez\textsuperscript{1} \quad Manolis Mavrikis\textsuperscript{2} \quad Darren Pearce\textsuperscript{1}

London Knowledge Lab
\textsuperscript{1}Birkbeck College \quad \textsuperscript{2}Institute of Education
23-29 Emerald Street, WC1N 3QS, London (UK)
E-mail: \{	exttt{sergut, manolis, darrenp}@lkl.ac.uk\}

Abstract

Although the notion of generality is central in mathematics and science, being able to identify and express general patterns and/or articulate structures is one of the main difficulties for children when they learn mathematics. This paper presents a step towards a set of tools that addresses this problem by encouraging students to connect between actual activities within the task and expressions of generality. The paper introduces a novel tool — ShapeBuilder — and describes how it can be used in the context of a well-known generalisation activity: tiling a pond. Insights gained from various small-scale exploratory studies with children are discussed followed by a description of how we expect this tool to develop in future work.

1. Introduction

The need to recognise, express and justify generality is at the core of mathematical thinking and scientific enquiry. However, a voluminous body of mathematics education research (see [9] for examples) suggests that expressing generality, recognising and analysing patterns, and articulating structure is complex and problematic for students. As suggested in [9], the difficulties students face should not be interpreted simply as their own failure but have to be investigated in the context of the curriculum, the nature of the tasks posed and the tools available for their solution.

In the traditional mathematical curriculum, algebra is a means of expressing generality. However, generalisation is so implicit in algebra that experts no longer notice the strategies they have integrated into their thinking [8]. This causes problems for students who perceive algebra as an endpoint rather than a tool for problem solving [9]. One of the reasons for this is that the questions asked and the actions to be taken have no real need for mathematical expression. The latter — even when achieved correctly — is often disconnected from the activity which precedes it which ‘neither illuminates the problem nor provides a means for validating its solution’ [5].

This paper presents a step towards a set of tools that addresses this problem by encouraging students to connect their actions within activities with the need to express generality. These tools are part of the MiGen Project\textsuperscript{1} which is developing an environment to promote the learning of mathematical generalisation.

The paper is structured as follows. Section 2 discusses related work from both the learning sciences and computer science fields. Section 3 briefly describes the learning environment. This is followed by Section 4 which details the insights obtained from testing this tool with various children on a specific example generalisation task using this environment. Section 5 closes the paper, drawing general conclusions and discussing future work.

2. Related Work

The difficulty that algebraic thinking poses to children has been thoroughly studied in the field of mathematics education [7, 3]. There have been numerous attempts to foster a shift towards an appreciation of generality, most notably through finding ways for students to construct their own mathematical models [9]. This modelling approach seeks not only to foster seeing the general in the particular by construction and exploration, but also a sense of ownership of the abstraction process. Despite some successes, difficulties remain, and these tend to coalesce around the need for significant pedagogic support from the teacher.

Our aim is to develop tools that provide assistance to learners and advice to teachers based on analyses of individual students and the activities of the group overall. De-

\textsuperscript{1}See \url{http://www.migen.org/} for more details. Funded by the TLRP, e-Learning Phase-II; grant number: RES-139-25-0381
spite the existence of some intelligent tutoring systems that attempt to provide support for students [6, 4], they are only suitable for children who already have a clear notion of algebraic notation and variables. With only few exceptions (e.g. [1]), they do not promote exploratory learning.

3. ShapeBuilder

ShapeBuilder is an environment which aims to encourage structured algebraic reasoning of 11-14 year-old students. It allows the learner to create constants, variables and arbitrarily complex compound expressions involving the operators addition, subtraction, multiplication and division. These expressions can then be used to define the width and height of rectangles. Once a shape is defined, the user can move it, attach it to other shapes and alter its expressions as desired. At any time, the current value of a variable can be edited directly. This can lead to an unlimited number of changes in the dimensions of the shapes currently defined.

As described so far, ShapeBuilder provides facilities for the learner to define shapes using expressions. However, a critical feature of the software allows users to define expressions using shapes. Specifically, by double-clicking on an edge of a rectangle, the user is able to obtain an iconic variable which, at all times, evaluates to the current value of that dimension of the rectangle. The iconic variables can be combined to form expressions which can be used to construct new shapes as shown in Figure 2d.

4. Using ShapeBuilder for Pond Tiling

Pond tiling is a very typical generalisation task for children in our target age group. Given a rectangular pond with a particular integer width and height, the task entails challenging the child to determine how many tiles are required to surround it. Although simple, this task is surprisingly rich in that it lends itself to a variety of different solutions each corresponding to a different general algebraic expression (Figure 1). Having constructed one of these possible solutions, the user can then change the values of the variables and observe their resulting tiling thus validating the generality of their solution across various specific cases.

The scope for alternative solutions allows the learner to realize how multiple valid solutions to a problem lead to different — but equivalent — expressions. There is then an incentive to develop some of the basic rules of algebra intuitively such as commutativity and associativity as well as provocation to collaborate with other children who have found correct but different solutions.

As part of a user-centred iterative design process, we have carried out several small-scale exploratory studies with various versions of the software in order to refine aspects of the user interface and investigate approaches students take to solving this tiling task. After a brief software familiarisation session, groups of two or three students are presented with the pond tiling activity and asked by the teacher/researcher to develop an expression that represents a rule that calculates the number of tiles needed to surround a pond.

One typical yet erroneous approach that students take is to construct their solution for a specific pond. An example of this is shown in Figure 2a where the width of the rectangle is defined by the constant 5. This looks correct for a pond of width 3 (Figure 2b). However, if the width of the pond is changed by the teacher (or indeed another student or, in the future, the system itself), the construction is no longer valid (Figure 2c).

In the dynamic geometry literature, this operation is referred to as ‘messing up’ [2]. The challenge for students therefore is to construct a solution that is impervious to ‘messing up’ in this way. ShapeBuilder lends itself to engaging with this interesting pedagogical approach by giving the student the opportunity to build the specific (as above) or the general through using iconic variables. This is illustrated in Figure 2d where the student has defined their rectangle in terms of the width of the pond. When the pond is resized, the tiling changes accordingly (Figure 2f). This ‘incentive to generalise’ [8] provides students with the opportunity to realize that there is an advantage to using iconic variables and facilitates thinking in terms of abstract characteristics of the task rather than specific measurements.

5. Conclusions and Future Work

Algebra in general, and the development of mathematical patterns in particular, is one of the more important difficulties in the learning of mathematics at an early age. We
have presented a tool that encourages the user to connect coherently between a concrete activity (pond tiling) and the actual algebraic expression that describes the general solution to the problem. Through some exploratory small-scale studies, we have seen that students are able to construct general solutions and describe them verbally. This facilitates the construction of general expressions or rules that represent the number of tiles around a pond.

Immediate progress of our research aims to provide intelligent feedback on the activity by the tool itself. This will help students in self-learning and distance learning scenarios as well as act as an effective supplement to the support that a teacher provides in the classroom. Since the actions of the students will be logged by the system as they happen, off-line as well as real-time analysis and feedback is possible. There are two important challenges. Firstly, the exploratory nature of the task we are dealing with means that there is a broad grey area between “clearly correct” and “clearly incorrect” that cannot be easily assessed. Some techniques have been used in the past for dealing with exploratory environments like Bayesian networks [1], combinations of heuristics and formal methods [10] or multi-agent systems [11]. Secondly, the domain of mathematical generalisation is vague and difficult to formalise. Our intention therefore is to develop ontologies to categorise specific task domains and possible strategies. In this way, we aim to develop a generic system that enables effective and meaningful interaction with a wide range of generalisation tasks.

References


