

Hiding the *Mathematica* and showing the mathematics

G. H. Smith, L.N. Wood

University of Technology, Sydney
gsmith@maths.uts.edu.au, leigh@maths.uts.edu.au

N. A. Nicorovici

The University of Sydney
nicolae@physics.usyd.edu.au

This paper describes an innovative contribution to *flexible learning*, using *Mathematica* in an interactive package¹. Unlike previous approaches which have used *Mathematica*, there is no need for the student to learn *Mathematica*, a process which may often obscure the mathematics. Our approach renders the *Mathematica* completely transparent by way of pre-programming, hyperlinks and buttons. The package takes advantage of *Mathematica* as the computational engine without the student needing to know any *Mathematica*, making it accessible to beginning university students. We have developed a teaching package in the context of discrete mathematics, but the ideas are applicable to other areas of mathematics.

Introduction

It is almost universally the case that teaching materials using *Mathematica* require the student to have a working knowledge of *Mathematica*. This project is unusual in that the *Mathematica* component is completely opaque to the student. The student requires only basic computer skills and the ability to use standard tools such as hyperlinks and buttons. Rather than requiring students to master the syntax of *Mathematica*, we have pre-programmed all problems and material, so that students are able to use graphics and interactive processes without the barrier of learning a programming language in their first semester of study. *Mathematica* is used as the computational engine and is kept in the background. Progression through the various concepts and the generation of problems for the student is by way of hyperlinks and buttons. In this way it is possible to concentrate on the concepts and provide a powerful learning tool.

One of the advantages of using *Mathematica* is that it is now the standard symbolic mathematics tool for many universities, including (in the Sydney region) UTS, the University of Sydney and Wollongong University. Various applications and packages are available over the World Wide Web and large numbers of students have their own copy of *Mathematica*. Students are expected to become familiar with its use as a learning tool as they progress through their studies. It is readily available, portable and integrated into many teaching programs.

Development is completed on two modules—one on set theory and one on functions. We are nearing completion on a further two modules, one on logic and one on graph theory. Student trials of the first two modules will begin in the Spring Semester of 1999. The notebooks are available on the university computer system and on floppy disks for home use.

¹This project was funded by a 1998 National Teaching Development Grant, Project Number 3016/3

Context

Discrete mathematics is an area of rapidly growing importance with applications in computing, telecommunications, engineering and the social sciences. The language and terminology used is very different from the more usual calculus and algebra courses that are encountered at school or in the initial years of university. There is a much larger emphasis on concepts and abstract ideas and many students find the transition difficult. Large numbers of students are now required to learn discrete mathematics, either as part of a subject or as a separate subject. Our aim is to assist students in concept development and to address misconceptions arising in the area of discrete mathematics, rather than the development of skills in *Mathematica*. In any case, the vast depth of *Mathematica* means that only a small part of its full potential is available to beginning students. Mastering the syntax to plot a complex graphical image with coloured infills, such as a Venn diagram, is rather difficult. In our case, the process is already pre-programmed for the students, so that they can use graphics and interactive processes without having to learn a programming language in their first semester of study.

The output consists of interconnected *Mathematica* notebooks on aspects of discrete mathematics, together with accompanying print materials. The notebooks explain the concepts and definitions and provide examples and exercises for the student. Hyperlinks enable the student to move through the notebooks to check concepts and definitions. We have taken advantage of the word processing capabilities of *Mathematica* to produce a user-friendly appearance and we have removed some of the more obvious *Mathematica* appendages such as cell brackets.

Target groups

The target groups are students majoring in mathematics, computing, electrical engineering, telecommunications and associated disciplines. The target learning styles within these groups are (i) visual learners and people who seek concrete illustrations of abstract concepts, (ii) non-English speaking background students who have difficulty with formal language, (iii) students from non-academic backgrounds who find the concepts and formality of the language threatening and (iv) students who like to use exploratory learning or who prefer a variety of different approaches to a concept in order to come to a full understanding of it. The learning outcomes we expect are the removal of misconceptions, deeper learning of concepts, knowledge of notations and vocabulary, visualisation of concepts, connections between different representations of the same concept, confidence in understanding new concepts, applications in new areas and, finally, excitement and enthusiasm for new ideas.

Literature and database review

A number of common misconceptions amongst students of mathematics have been identified by Tall [5], who summarises the literature in the area and quotes a study by Dreyfus and Vinner who asked a cross-section of students and teachers a number of conceptual questions about functions. The study showed that the reason for many of the incorrect responses was that students often had very clear but mistaken ideas as to precisely what constituted a mathematical function. Addressing such misconceptions has been an important part of our project

There are many resources available on the World Wide Web, but these are often stand-alone

resources without the flexibility, power and integration of the project we are describing (for example Venn Diagrams² and a Propositional logic program³). Our project, which is based on *Mathematica*, is capable of improvement and expansion in line with future developments and improvements in *Mathematica*. There are teaching and learning aids using *Mathematica* as a basis ([1], [2], [3], [4]), but these do not contain the adaptations that our project uses to make the resource available to beginning students. Some Web resources which use *Mathematica* emphasise technical skills rather than concepts⁴, while others such as Courseware⁵, from The University of Toronto assume student knowledge well above that of our target groups and keep the *Mathematica* syntax in the foreground, something which we have consistently avoided.

Examples

We illustrate some of the ideas of the package by means of specific examples.

1. **Venn diagrams:** One of our intentions is to help students make links between different representations of the same concept—in this case between visual and symbolic representations. The relevant part of the *Mathematica* notebook appears in Figure 1 below.

In this task, you will be presented with a random Venn Diagram and asked to type in an equivalent set-theoretic expression. Click the button to begin.

Figure 1: An exercise on Venn diagrams

When the student clicks on the button, *Mathematica* randomly generates a Venn diagram such as the one shown in Figure 2. The student is asked to express the coloured region in set theoretic notation.

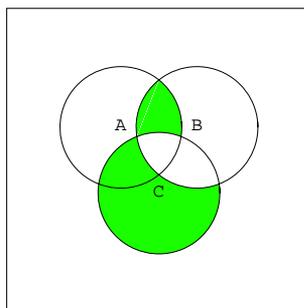


Figure 2: Venn diagram

The student's answer is entered via the keyboard together with the use of a symbol palette. *Mathematica* then represents the answer graphically as a second Venn diagram. If the student has typed any one of the many logically correct expressions—such as $(C \setminus B) \cup ((A \cap B) \setminus C)$ or $((A \cap B) \cup C) \setminus (B \cap C)$ —then the two diagrams will be identical, so that the feedback is immediate. Even if the student enters an incorrect symbolic expression, he or she will have an immediate graphical representation of this wrong answer, which helps to provide understanding of why the answer is incorrect.

²<http://archives.math.utk.edu/software/msdos/discrete.math/venn/.html>

³<http://archives.math.utk.edu/software/msdos/discrete.math/jkpropc/.html>

⁴<http://CalMaeth.maths.uwa.edu/>

⁵<http://www.scar.utoronto.ca/homes/mathlab/courseware.html>

Here even an incorrect answer has provided a positive learning experience. This is a characteristic of much of our project—mistakes can be used positively. The two diagrams will match only when a correct answer is given.

To get some idea of the hidden *Mathematica* syntax, Figure 3 gives the full *Mathematica* syntax which generates the cell in Figure 1.

```
Cell[BoxData[
  ButtonBox[
    RowBox[{" ",
      RowBox[{"Generate", " ", "a", " ", "Random", " ", "Venn", " ", "Diagram"}]}],
    ButtonFunction:>(CompoundExpression[
      SelectionMove[
        ButtonNotebook[ ], After, ButtonCell],
      SelectionMove[
        ButtonNotebook[ ], Next, Cell],
      NotebookWrite[
        ButtonNotebook[ ],
        Cell[
          GraphicsData[ "PostScript",
            DisplayString[ #2]], ShowCellBracket -> False, Background -> None, CellFrame
            -> False]]&),
      ButtonEvaluator->Automatic,
      ButtonData:>RandomVennDiagram[ ],
      Active->True,
      ButtonStyle->"Evaluate"]], "Input",
    ShowCellBracket->False,
    GeneratedCell->True,
    Background->None]
```

Figure 3: The code behind the button

This is only part of the work required. There is a reference to `RandomVennDiagram[]` six lines from the bottom. This is the function which is called by the button to generate the random Venn diagram. This function is contained in a 490 line Venn diagram package which we have written for this project.

2. **Relations:** This is an exercise designed to develop routine skills. *Mathematica* is able to check the various answers that the student provides and to provide feedback on incorrect or partially correct answers.

Let $A = \{1, 2, 3, 4, 5\}$ and let $B = \{2, 4, 6, 8\}$. Write down relations R_1 , R_2 and R_3 from A into B such that:

- (a) if $(a, b) \in R_1$, then $a \leq 3$ and $b \geq 6$.
- (b) if $(a, b) \in R_2$, then a is even and b is divisible by 4.
- (c) if $(a, b) \in R_3$, then b is divisible by a .

Enter your answers in the form $\{(a_1, b_1), (a_2, b_2), (a_3, b_3), \dots, (a_n, b_n)\}$ in the templates for R_1 , R_2 and R_3 below and then click the evaluation button. Click the answer button to check your answers.

There are many answers to each part, so repeat the exercises by giving as many different answers as you can.

3. **Functions:** A typical exercise is one in which the student is given the domain and codomain of a function f and then asked to explicitly enter examples of f which satisfy certain properties. For example, the domain may be given as $\{a, b, c\}$, codomain as $\{t, u, v, w\}$ and the student requested to construct a one-to-one function with the given domain. If, for example, the student enters $\{(a, w), (b, u), (c, t)\}$, then a diagram such as Figure 4 appears and there is confirmation that the student's function is indeed one-to-one.

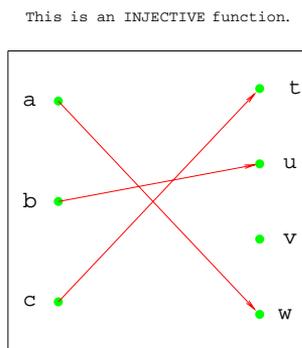


Figure 4: Function representation

Conclusion

We have developed materials which assist students to make connections between different representations of the same concept—verbal, graphical and algebraic. The use of visual self-correction gives immediate feedback. *Mathematica* has proved to be an excellent tool, owing to its computational capabilities and the principles which we have used are applicable in many mathematical areas. The down side is the large programming effort that is required, both in terms of funds and expertise.

References

- [1] Cohen, J.K.,(1992). Packages for Logic and Set Theory, *Mathematica Journal* 2, 91–93.
- [2] Maeder, R.E.,(1994). The Mathematica Programmer: Logic Programming I: The Interpreter, *Mathematica Journal* 4, 53–63.
- [3] Maeder, R.E.,(1994). Logic Programming II: Applications, *Mathematica Journal* 4, 38–43.
- [4] Murrell,K., (1992). Animation of Rotating Rigid Bodies, *Mathematica Journal* 2, 61–65.
- [5] Tall, D.O., (1992). *The transition to advanced mathematical thinking: functions, limits and proof*, in D.A.Grouws (ed.) *Handbook of Research on Mathematics Teaching and Learning*, Macmillan, New York, 495–511.