Proceedings of the International Symposium on Mathematical Practice and Cognition , Alison Pease, Markus Guhe, and Alan Smaill (Eds.),
at the AISB 2010 convention, 29 March – 1 April 2010, De Montfort University, Leicester, UK

Applying the GC Combined Reasoning Framework to Mathematical Discovery

John Charnley[1]

Abstract.

In previous work, we have demonstrated how the GC combined reasoning framework has been used for mathematical discovery. However, the current approach has several limitations, for example a single basic representation scheme. By reference to a mathematics text book, we highlight some challenging aspects of mathematical discovery and comment on how characteristics of the GC approach provide useful solutions. In particular, the distributed nature of the framework would allow multiple representation schemes to be considered, which is useful in human studies of mathematics and allows us to introduce more advanced reasoning systems. We believe that, by implementing these solutions, we can create a highly effective mathematical discovery package.

1 INTRODUCTION

Automatic mathematical discovery has long been a goal of AI researchers and several systems have been developed, for example Graffiti [5] and HR [3]. In general, these systems aim to create theories that would be both interesting to, and verifiable by, human mathematicians. The creation of a mathematical theory involves developing new mathematical concepts, identifying relationships between them and discovering and proving theorems. Consequently, it encompasses numerous specific skills such as symbolic manipulation, theorem proving, example generation etc. As a result, several previous systems, such as HR, employ a combination of automated reasoning techniques. In previous work, we developed the GC framework for combined reasoning, which we describe in §2. The GC framework, which takes its name from the initial letters of Global Workspace Theory and combined reasoning, allows users to create effective systems which employ multiple forms of automated reasoning in performing a particular task [2].

We have already used GC to create a discovery system, GCATF, which takes HR as its inspiration, and we have applied it to a number of mathematical domains. However, GCATF has several limitations and we believe there is a great deal of scope to extend its capabilities. In particular, the GC approach has characteristics which we believe will be of benefit in mathematical discovery tasks. The most obvious limitation is the system's representation scheme. Mathematicians use a variety of visualisations and representations for mathematical notions. Like the underlying ideas they represent, they vary in their complexity. Furthermore, the representations are dynamic, having developed hand-in-hand with the subject of mathematics itself. Most discovery systems use a limited representation scheme. In the case of GCATF, this is Prolog-style first-order logic. Whilst useful for many domains, this single static representation places an artificial ceiling on the complexity of mathematical notions that can be considered. It is also a barrier to leveraging more advanced automated reasoning systems.

In order to better understand this issue and other deficiencies of GCATF, we found it useful to go back to basics. So, we reviewed a typical mathematical textbook in the area of mathematical analysis [9]. In only the first few pages, we see several instances where a more complex approach is needed. In §3, we describe some of these and comment on how GC offers potential solutions.

2 The GC Framework

In cognitive science, Global Workspace Theory (GWT) has been proposed to describe the operation of mammalian consciousness [1]. At its heart lies the Global Workspace Architecture (GWA), which describes an arrangement where a number of parallel specialist processes are attached to a global workspace. In each round of processing, some information is broadcast on the workspace. In response, the attached processes can, if they wish, propose some new information for broadcast. The information proposed by the process making the 'strongest' proposal is broadcast in the next round of processing. The GC framework allows developers to create distributed systems according to GWT. In the GC setting, each of the processes performs some form of reasoning and the contents of the workspace are reasoning artefacts. In addition, the GWT is modified slightly to allow processes to spawn and attach new processes during processing. The GC toolkit provides software allowing users to develop processes, bring them together into combined systems and run them in either a serial or distributed manner. To do this, a user defines the processes to be included and the reasoning artefacts that may be broadcast. Overall control is implicit in how the processes weight their suggestions, so the user must specify an appropriate weighting system. The framework is extremely light-weight and GC provides only the skeleton with which to develop and run systems. An effective system requires the user to implement specialist reasoning processes, for example by creating wrappers for external stand-alone reasoning systems. Alternatively, they could re-use or adapt existing processes from previous GC systems.

GC offers several benefits for creating combined reasoning systems. The use of specialist processes and a simple communication scheme means that GC systems are modular, clear and extensible. In addition, the architecture addresses issues of relevancy by neatly distributing the choice of which action or tool to apply. Importantly, the architecture imposes no specific semantics or representation scheme to be followed. The GC framework has been shown as an effective basis for many combined reasoning applications [2]. In particular it

[1] Imperial College,London, email:john.charnley@imperial.ac.uk

lends itself to free-form discovery tasks, where specific sub-systems contribute to a task when they see themselves to be relevant.

Several other systems that have been developed with similar goals to those of the GC project are described in [2]. For instance, Sorge's Ω-ANTS framework [10] and the MATHWEB architecture [7]. These systems also aim to facilitate the inter-operation of heterogeneous distributed reasoning systems in mathematics. However, these systems have mainly been designed for a particular purpose, such as automated theorem proving, and their suitability for more free-form discovery tasks has not been established.

3 Textbook Mathematics

Mathematics text books offer one way of understanding the discovery process from a human perspective. We read the opening chapters of an elementary mathematics text book [9], which is a first year degree-level introduction to analysis. It is interesting in that it aims to build the infinitesimal calculus from the basic axioms of real numbers. As such, its structure is similar to how we would want our automated discovery system to create theorems from first principles. Although this is a very basic starting point, the first few pages of this book highlight a number of key considerations, including:

- **Representation**: the book begins with some initial mathematical concepts which are considered known to the reader. For example, the concept of a real number is not formally defined but taken from the reader's knowledge of continuous measurements, such as mass. A common human understanding of real numbers is a mental visualisation as a line, together with certain allowed manipulations and expected behaviours. Using analogies in this way is key aspect of human mathematics [8]. In addition, there are also symbolic representations for concepts. Indeed, there are often several visualisations or symbolic representations. For example, $sign(x)$ can be defined as $-1 \leftrightarrow x < 0$ and $1 \leftrightarrow x >= 0$, alternatively it can be seen as $x/|x|$, both of which are useful. The ability to move between different representations of the same concept is fundamental in mathematical discovery, for instance in applying varied proving techniques. Consequently, an effective mathematical discovery system should have access to multiple representations.

  GC provides one answer to this problem by allowing different conceptual interpretations to be distributed. We refer to this as *distributed representation*. At the simple level, this involves using sub-systems with different formalisms and making the interpretation task local to those sub-systems. At a more complex and challenging level, we envisage leveraging the parallel nature of GC to consider all possible representations for a given concept in a given context as required. The GC framework has its origin in cognitive science, which may provide insights into how humans employ different schemes. For instance, we may be able to mimic the way people identify loose similarities between different ideas, leading to the development of concrete relationships.

- **Concept Formation**: the book introduces new concepts in several ways. Many of the mechanisms used are similar to those we have considered in previous studies. For example, concepts are introduced as being particular properties of existing concepts e.g. $|x|$. However, some methods go beyond the scope of our existing methods, for example concepts which involve infinity, e.g. infinite sequences and series. In addition, different representations require their own specific concept formation methodologies, e.g. symbolic transformation or diagrammatic manipulation. Consider, for instance, the development of the concept of a plane as two perpendicular real lines versus its symbolic analogy of bringing two real numbers into a tuple, (x, y). Again, this is challenging, especially when trying to automatically ensure consistency between representations. It is unlikely that analogues of all concepts could be maintained in every representation scheme at all stages. This would require the system to create, for each new concept generated in one representation, a semantically equivalent concept in all other representations, which may not be possible. Perhaps a more achievable approach would be to allow concept formation to continue freely in multiple representation schemes. The identification of analogues would then become a separate task, which might sporadically identify correspondences.

  One benefit of the GC approach is the ease with which it is possible to add new functionality. This feature should allow us to incrementally boost our concept formation capabilities as we introduce new formation methods or develop approaches to maintaining consistency of representation.

- **Investigative Direction**: the book progresses by introducing new interesting concepts and investigating them in some depth, with new theorems and exercises. In our system, concept formation drives the direction of the investigation. However, in a free-form discovery task, often a large body of uninteresting information obscures the important discoveries and focus can be lost. Some work has already been done on assessing interestingness of generated concepts, for example by using feature analysis to rank them [4].

  GC's weighting system provides a mechanism by which we might introduce interestingness and control the direction of investigation. It would also be useful to make this dynamic by introducing a retrospective review of how important a particular concept becomes as the discovery process progresses.

- **Symbolic Manipulation and Proving**: this key aspect of mathematical study is, fortunately, a very successful area of AI and has spawned many different systems and approaches, including computer algebra systems and provers for many types of logics. To date we have only used a tiny fraction of the available systems and there is vast scope to leverage advances in the field. This is particularly relevant if we are to use multiple representations. For instance, the first exercises in the textbook ask for a proofs of the irrationality of $\sqrt{2}$ and $\sqrt{3}$. We were able to prove both of these, the first using purely symbolic manipulation but we required an element of diagrammatic reasoning to convince ourselves of the second. There are existing systems for each of these approaches which we would seek to use, for example the DIAMOND diagrammatic reasoning program [6].

  The GC approach engenders extensibility, meaning that introducing new tools is straightforward. However, one problem is that it is not always possible to leverage the full power of external reasoning systems as they often rely upon some form of human intervention to achieve their full potential. Adding new systems hopefully gives us an opportunity to feed back into the development of external reasoning tools by giving an assessment of their usability by non-expert users and providing a common test platform.

4 SUMMARY

GC has already been shown to be a useful tool for mathematical discovery but there is a great potential for improvement. Several aspects of the GC approach suggest it would be able to deal with specific challenges that mathematical discovery presents. In particular, we believe the way in which GC distributes relevancy allows us to introduce multiple representation schemes. The extensibility of GC systems means it is relatively straightforward to introduce new functionality, such as advanced concept formation methods or more powerful reasoning tools. Moreover, the weighting scheme provides a way in which we might introduce notions of interestingness by which to guide the discovery process. We believe that GC represents a promising basis for a mathematical discovery package which would provide an opportunity to bring together disparate reasoning systems and contribute to their development.

ACKNOWLEDGEMENTS

REFERENCES

[1] B Baars, *In the theater of consciousness: The workspace of the mind*, Oxford University Press, 1997.
[2] J Charnley, *A Global Workspace Framework for Combined Reasoning*, Ph.D. dissertation, Imperial College, London, 2010.
[3] S Colton, *Automated Theory Formation in Pure Mathematics*, Springer-Verlag, 2002.
[4] S Colton, A Bundy, and T Walsh, 'On the notion of interestingness in automated mathematical discovery', *International Journal of Human Computer Studies*, **53(3)**, 351–375, (2000).
[5] S Fajtlowicz, 'On conjectures of graffiti', *Discrete Mathematics*, **72**, 23,113–118, (1988).
[6] Mateja Jamnik, *On Automating Diagrammatic Proofs of Arithmetic Arguments*, Ph.D. dissertation, Division of Informatics, University of Edinburgh, 1999.
[7] M Kohlhase and J Zimmer, 'System description: The MathWeb software bus for distributed mathmatical reasoning', in *Proceedings of the Conference on Automated Deduction (CADE)*, (2002).
[8] George Lakoff and Rafael E. Núñez, *Where Mathematics Comes From: How the Embodied Mind Brings Mathematics into Being*, Basic Books, 2000.
[9] John B. Reade, *An Introduction to Mathematical Analysis*, Oxford Science Publications, 1986.
[10] V Sorge, *A Blackboard Architecture for the Integration of Reasoning Techniques into Proof Planning*, Ph.D. dissertation, Universität des Saarlandes, Saarbucken, 2001.